IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT APPLICATION

Appellants	:	Wilson Wong et al.
Application No.	:	10/762,864 Confirmation No. : 5928
Filed	:	January 21, 2004
For	:	ADAPTIVE EQUALIZATION METHODS AND APPARATUS FOR PROGRAMMABLE LOGIC DEVICES
Art Unit	:	2611
Examiner	:	Aristocratis Fotakis

New York, New York 10036
January 13, 2009

Mail Stop APPEAL BRIEFS - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

APPEAL BRIEF UNDER 37 C.F.R. § 41.37

Sir:

Appellants are filing this Appeal Brief in support of their appeal from the final rejection of claims 1, 3-8, 10-12, 14-16, 20-22, 24, 26-28, 30 and 32-44 in the final Office Action dated October 15, 2009 ("Office Action"). A Notice of Appeal was filed in this application on November 16, 2009.

In view of the arguments and authorities set forth below, the Board should find the rejections of claims 1, 3-8,

10-12, 14-16, 20-22, 24, 26-28, 30 and 32-44 to be in error, and the Board should reverse those rejections.

This Brief has the following appendices:

Claims Appendix

Appendix A: Copy of claims 1, 3-8, 10-12, 14-16, 20-22, 24, 26-28, 30 and 32-44 involved in this appeal;

Evidence Appendices

Appendix B: Copy of the final Office Action dated October 15, 2009;

Appendix C: Copy of U.S. Patent Application Publication No. 2004/0071205 ("Gorecki");

Appendix D: Copy of WinSLAC Software User's Guide (1999) ("WinSLAC");

Appendix E: Copy of Altera Corporation, "FIR Compiler MegaCore Function" ("Solution Brief 41");

Appendix F: Copy of U.S. Patent Application Publication No. 2005/0047779 ("Jaynes"); and

Appendix G: Copy of Hillery U.S. Patent No. 6,178,201 ("Hillery")

Appendix H: Copy of Lu U.S. Patent No. 6,275,836 ("Lu")

Appendix I: Pedersen et al. U.S. Patent Application Publication No. 2006/0114979 ("Pedersen")

Related Proceedings Appendix

None.

(i) REAL PARTY IN INTEREST

Appellants respectfully advise the Board that the real party in interest in the above-identified patent application Altera Corporation, a corporation organized and existing under the laws of the State of Delaware, and having an office and place of business at 101 Innovation Drive, San Jose, California 95134-1941, which is the assignee of the above-identified patent application.

(ii) RELATED APPEALS AND INTERFERENCES

Appellants respectfully advise the Board that there are no other appeals or interferences known to appellants, their legal representative, or their assignee, that will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(iii) STATUS OF CLAIMS

Claims 1, 3-8, 10-12, 14-16, 20-22, 24, 26-28, 30 and 32-44, which are the only claims pending in the above-identified patent application, stand finally rejected in this application and are the claims that are the subject of this appeal.

(iv) STATUS OF AMENDMENTS

Appellants have not submitted any amendments pursuant to 37 C.F.R. § 1.116, or otherwise, subsequent to the Office Action, from which this appeal is being taken.

(v) SUMMARY OF CLAIMED SUBJECT MATTER

Appellants' independent claims 1, 8, 12, 22, 24, 26, 27, 28, 30 and 32, are directed to receiver circuitry and methods for adaptively equalizing a data signal. The receiver circuitry and methods include, inter alia, programmable circuitry that is programmed with a first value (claims 1, 8, 12, 24, 26, 27, 28, and 32) or with a first training pattern (claims 22 and 30). The programmable circuitry outputs the first value (claims 12, 24, 28, and 32) or the first training pattern (claims 22 and 30) and a control signal. Processing circuitry computes a second value (claims 1, 8, 12, 24, 26, 27, 28, and 32) and outputs the second value in parallel with the first value (claims 12, 24, 28, and 32). Training pattern circuitry stores a second training pattern and outputs the second training pattern in parallel with the first training pattern (claims 22 and 30). Selection circuitry selects, based on a control signal (claims 1, 12, 22, 24, 27, 28, 30, and 32), one of the first and the second values (claims 1, 8, 12, 24, 26, 27, 28, and 32) or one of the first and second training patterns (claims 22 and 30) at the time the programmable circuitry is being programmed by configuration data.

Claims 1 and 26 recite that the first and second values each correspond to a "number of taps" of a filter. Claims 8 and 27 recite that the first and second values each indicate a "selection between integer spacing and fractional spacing" of filter taps. Claims 12 and 28 recite that the first and second values each correspond to a "starting value." Claims 24 and 32 recite that the first and second values each correspond to a "location of the sampling point."

Support in the specification for claims 1, 8, 12, 22, 24, 26, 27, 28, 30 and 32 is found at least in the locations indicated in the following table:

Claim 1	The Specification
1. Receiver circuitry for adaptively equalizing a data signal, the receiver circuitry comprising:	*See*, e.*g.*, page 4, line 11 to page 5, line 2, FIGS. 1, 3-5, receiver circuitry 30.
equalization implementation circuitry that includes a selectable number of taps, wherein the equalization implementation circuitry operates on the data signal;	*See*, e.*g.*, page 5, lines 3 to 20, page 6, line 12 to page 7, line 8, and FIG. 3, circuitry 40 and element 110.
programmable circuitry that is programmed by configuration data with a first value corresponding to a first number of taps;	*See*, e.*g.*, page 7, line 9 to page 8, line 8, and FIG. 3, programmable elements 116.
processing circuitry that computes a second value corresponding to a second number of taps; and	*See*, e.*g.*, page 7, line 9 to page 8, line 8, page 8, lines 21 to 29, and FIG. 3, element 118.
selection circuitry that selects, based on a control signal, one of the first and second values as the selectable number of taps at the time the programmable circuitry is being programmed by the configuration data, wherein the selection circuitry selects one of the first and second values only once while the equalization implementation circuitry operates on the data signal; and	*See*, e.*g.*, page 7, line 9 to page 8, line 8, and FIG. 3, multiplexer circuitry 112, element 114.

Claim 1	The Specification
wherein the selectable number of taps of the equalization implementation circuitry corresponds to the selected one of the first and second values.	*See, e.g.*, page 7, line 9 to page 8, line 8.

Claim 8	The Specification
8. Receiver circuitry for adaptively equalizing a data signal, the receiver circuitry comprising:	*See, e.g.*, page 4, line 11 to page 5, line 2, FIGS. 1, 3-5, receiver circuitry 30.
equalization implementation circuitry that includes a filter with taps having a selected one of integer tap spacing and fractional tap spacing relative to the symbol rate of the data signal, wherein the equalization implementation circuitry operates on the data signal;	*See, e.g.*, page 5, lines 3 to 20, page 9, lines 14 to 21, and FIG. 3, circuitry 40 and element 120.
programmable circuitry that is programmed by configuration data with a first value indicating a first selection between integer spacing and fractional spacing of the taps;	*See, e.g.*, page 9, lines 22 to 33, and FIG. 3, programmable elements 126.
processing circuitry that computes a second value indicating a second selection between integer spacing and fractional spacing of the taps; and	*See, e.g.*, page 9, lines 22 to 33, page 10, lines 1 to 8 and FIG. 3, element 128.
selection circuitry that selects one of the first and second values as the	*See, e.g.*, page 9, lines 22 to 33, and FIG. 3, multiplexer circuitry 122.

Claim 8	The Specification
selected one of integer spacing and fractional spacing at the time the programmable circuitry is being programmed by the configuration data, wherein the selection circuitry selects one of the first and second values only once while the equalization implementation circuitry operates on the data signal; and	
wherein the selection between the integer and fractional tap spacing corresponds to the selected one of the first and second values.	*See*, e.*g.*, page 9, lines 22 to 33.

Claim 12	The Specification
12. Receiver circuitry for adaptively equalizing a data signal, the receiver circuitry comprising:	*See*, e.*g.*, page 4, line 11 to page 5, line 2, FIGS. 1, 3-5, receiver circuitry 30.
equalization implementation circuitry that includes at least one selectable coefficient value;	*See*, e.*g.*, page 5, lines 3 to 20, page 10, lines 18 to 32, and FIG. 4, circuitry 40 and element 130.
first processing circuitry for computing the coefficient value using a selectable starting value, wherein the coefficient value is different from the starting value;	*See*, e.*g.*, page 12, lines 17 to 32 and FIG. 4, element 132.
programmable circuitry that is programmed by configuration data with a first starting value and outputs the first starting	*See*, e.*g.*, page 10, lines 18 to 32, and FIG. 4, programmable elements 138, element 136.

Claim 12	The Specification
value and a control signal;	
second processing circuitry that computes a second starting value and outputs the second starting value in parallel with the first value; and	*See, e.g.*, page 10, line 18 to page 11, line 18 and FIG. 4, element 140.
selection circuitry that: receives the control signal from the programmable circuitry, the first starting value and the second starting value in parallel;	*See, e.g.*, page 10, lines 18 to 32, and FIG. 4, multiplexer circuitry 134.
selects, based on the control signal, one of the first and second starting values at the time the programmable circuitry is being programmed by the configuration data, wherein the selection circuitry selects one of the first and second starting values only once; and	*See, e.g.*, page 10, lines 18 to 32, and FIG. 4, multiplexer circuitry 134.
outputs the selected one of the first and second starting values to the first processing circuitry,	*See, e.g.*, page 10, lines 18 to 32, and FIG. 4, multiplexer circuitry 134.
wherein the selectable starting value of the first processing circuitry corresponds to the selected one of the first and second values.	*See, e.g.*, page 10, lines 18 to 32.

Claim 22	The Specification
22. Receiver circuitry for adaptively equalizing a received data signal, the receiver circuitry	*See, e.g.*, page 4, line 11 to page 5, line 2, FIGS. 1, 3-5, receiver circuitry 30.

Claim 22	The Specification
comprising:	
first processing circuitry for computing a first error signal using a selectable training pattern, wherein the first processing circuitry operates on the data signal;	*See*, e.*g.*, page 5, lines 3 to 20, page 11, lines 19 to 29, and FIG. 4, circuitry 40 and algorithm 166.
programmable circuitry that is programmed by configuration data with a first training pattern and outputs the first training pattern and a first control signal;	*See*, e.*g.*, page 12, lines 5 to 16, and FIG. 4, programmable elements 174, programmable element 172.
training pattern circuitry that stores a second training pattern and outputs the second training pattern in parallel with the first training pattern; and	*See*, e.*g.*, page 12, lines 5 to 16, and FIG. 4, element 176.
first selection circuitry that:	*See*, e.*g.*, page 12, lines 5 to 16, and FIG. 4, multiplexer circuitry 170.
receives the first control signal from the programmable circuitry, the first training pattern and the second training pattern in parallel;	*See*, e.*g.*, page 12, lines 5 to 16, and FIG. 4, multiplexer circuitry 170.
selects, based on the first control signal, one of the first and second training patterns at the time the programmable circuitry is being programmed by the configuration data, wherein the first selection circuitry selects one of the first and second training patterns only once while	*See*, e.*g.*, page 12, lines 5 to 16, and FIG. 4, multiplexer circuitry 170.

Claim 22	The Specification
the processing circuitry operates on the data signal; and	
outputs the selected one of the first and second training patterns to the first processing circuitry.	*See, e.g.*, page 11, lines 19 to 29 and page 12, lines 5 to 16.

Claim 24	The Specification
24. Receiver circuitry for adaptively equalizing a data signal, the receiver circuitry comprising:	*See, e.g.*, page 4, line 11 to page 5, line 2, FIGS. 1, 3-5, receiver circuitry 30.
equalization implementation circuitry, in the receiver circuitry, having at least one sampling point with a selectable location relative to a bit period of the received signal, wherein the equalization implementation circuitry operates on the data signal;	*See, e.g.*, page 5, lines 3 to 20, page 13, lines 6 to 27, and FIG. 5, circuitry 40 and element 180.
programmable circuitry that is programmed by configuration data with a first value corresponding to a first location of the sampling point and outputs the first value and a control signal;	*See, e.g.*, page 13, lines 6 to 27, and FIG. 5, programmable elements 186, element 184.
processing circuitry that computes a second value corresponding to a second location of the sampling point and outputs the second value in parallel with the first value; and	*See, e.g.*, page 13, lines 1 to 5, page 13, lines 6 to 27, and FIG. 5, element 188.
selection circuitry that:	
receives the control	*See, e.g.*, page 13, lines

Claim 24	The Specification
signal from the programmable circuitry, the first value and the second value in parallel;	6 to 27, and FIG. 5, multiplexer circuitry 182.
selects, based on the control signal, one of the first and second values at the time the programmable circuitry is being programmed by the configuration data, wherein the selection circuitry selects one of the first and second values only once, while the equalization implementation circuitry operates on the data signal; and	*See, e.g.*, page 13, lines 6 to 27, and FIG. 5, multiplexer circuitry 182.
outputs the selected one of the first and second values to the equalization implementation circuitry,	*See, e.g.*, page 13, lines 6 to 27, and FIG. 5, multiplexer circuitry 182.
wherein the location of the at least one sampling point of the equalization implementation circuitry corresponds to the selected one of the first and second values.	*See, e.g.*, page 13, lines 6 to 27.

Claim 26	The Specification
26. A method of operating receiver circuitry having programmable circuitry and adaptive equalization capability, the method comprising:	*See, e.g.*, page 4, line 11 to page 5, line 2, FIGS. 1, 3-5, receiver circuitry 30.
programming the programmable circuitry using configuration data with a first value corresponding to a first number of taps;	*See, e.g.*, page 7, line 9 to page 8, line 8, and FIG. 3, programmable elements 116.

Claim 26	The Specification
computing a second value corresponding to a second number of taps;	*See*, e.*g.*, page 7, line 9 to page 8, line 8, page 8, lines 21 to 29, and FIG. 3, element 118.
selecting, at the time the programmable circuitry is being programmed by the configuration data, one of the first and second values, wherein the selecting selects one of the first and second values only once while equalization implementation circuitry operates on a data signal;	*See*, e.*g.*, page 7, line 9 to page 8, line 8, and FIG. 3, multiplexer circuitry 112. *See*, e.*g.*, page 5, lines 3 to 20, page 6, line 12 to page 7, line 8, and FIG. 3, circuitry 40 and element 110.
providing the selected one of the first and second values to the equalization implementation circuitry; and	*See*, e.*g.*, page 7, line 9 to page 8, line 8, and FIG. 3, multiplexer circuitry 112.
controlling the equalization implementation circuitry to operate with a number of taps corresponding to the selected one of the first and second values.	*See*, e.*g.*, page 7, line 9 to page 8, line 8 and FIG. 3, element 114.

Claim 27	The Specification
27. A method of operating receiver circuitry having programmable circuitry and equalization implementation circuitry that includes a filter with taps, the method comprising:	*See*, e.*g.*, page 4, line 11 to page 5, line 2, FIGS. 1, 3-5, receiver circuitry 30. *See*, e.*g.*, page 5, lines 3 to 20, page 9, lines 14 to 21, and FIG. 3, circuitry 40 and element 120.
programming the programmable circuitry using configuration data with a first value indicating a first selection between integer	*See*, e.*g.*, page 9, lines 22 to 33, and FIG. 3, programmable elements 126.

Claim 27	The Specification
spacing and fractional spacing of the taps;	
computing a second value indicating a second selection between integer spacing and fractional spacing of the taps;	*See, e.g.*, page 9, lines 22 to 33, page 10, lines 1 to 8 and FIG. 3, element 128.
selecting, at the time the programmable circuitry is being programmed by the configuration data, based on the received control signal, one of the received first and second values, wherein the selecting selects one of the received first and second values only once while the equalization implementation circuitry operates on a data signal;	*See, e.g.*, page 9, lines 22 to 33, and FIG. 3, multiplexer circuitry 122.
providing the selected one of the first and second values to the equalization implementation circuitry; and	*See, e.g.*, page 9, lines 22 to 33, and FIG. 3, multiplexer circuitry 122.
controlling the filter of the equalization implementation circuitry, in the receiver circuitry, to operate with the tap spacing corresponding to the selected one of the first and second values.	*See, e.g.*, page 9, lines 22 to 33.

Claim 28	The Specification
28. A method of operating receiver circuitry having programmable circuitry and adaptive equalization capability, the method comprising:	*See, e.g.*, page 4, line 11 to page 5, line 2, FIGS. 1, 3-5, receiver circuitry 30. *See, e.g.*, page 5, lines 3 to 20, page 10, lines 18 to 32, and FIG. 4, circuitry 40 and element 130.

Claim 28	The Specification
programming the programmable circuitry using configuration data with a first starting value, wherein the programmable circuitry outputs the first starting value and a control signal;	*See*, e.*g.*, page 10, lines 18 to 32, and FIG. 4, programmable elements 138, element 136.
computing a second starting value;	*See*, e.*g.*, page 10, line 18 to page 11, line 18 and FIG. 4, element 140.
receiving the control signal, the first starting value and the second starting value in parallel;	*See*, e.*g.*, page 10, lines 18 to 32, and FIG. 4, multiplexer circuitry 134.
selecting, at the time the programmable circuitry is being programmed by the configuration data, based on the received control signal, one of the received first and second starting values, wherein the selecting selects one of the received first and second values only once while equalization implementation circuitry operates on a data signal;	*See*, e.*g.*, page 10, lines 18 to 32, and FIG. 4, multiplexer circuitry 134.
processing the selected one of the first and second starting values to compute a coefficient value different from the selected starting value; and	*See*, e.*g.*, page 12, lines 17 to 32 and FIG. 4, element 132.
operating the equalization implementation circuitry, in the receiver circuitry, using the computed coefficient.	*See*, e.*g.*, page 5, lines 3 to 20, page 10, lines 18 to 32, and FIG. 4, circuitry 40 and element 130.

Claim 30	The Specification
30. A method of operating receiver circuitry having programmable circuitry and adaptive equalization capability, the method comprising:	See, e.g., page 4, line 11 to page 5, line 2, FIGS. 1, 3-5, receiver circuitry 30.
programming the programmable circuitry using configuration data with a first training pattern, wherein the programmable circuitry outputs the first training pattern and a first control signal;	See, e.g., page 12, lines 5 to 16, and FIG. 4, programmable elements 174, programmable element 172.
computing a second training pattern to output the second training pattern;	See, e.g., page 12, lines 5 to 16, and FIG. 4, element 176.
receiving the first control signal, the first starting training pattern and the second training pattern in parallel; and	See, e.g., page 12, lines 5 to 16, and FIG. 4, multiplexer circuitry 170.
selecting, based on the first control signal, one of the received first and second training patterns at the time the programmable circuitry is being programmed by the configuration data, wherein the selecting selects one of the first and second the training patterns only once, based on the configuration data, while equalization implementation circuitry operates on a data signal.	See, e.g., page 12, lines 5 to 16, and FIG. 4, multiplexer circuitry 170.

Claim 32	The Specification
32. A method of operating receiver circuitry having	See, e.g., page 4, line 11 to page 5, line 2, FIGS.

Claim 32	The Specification
programmable circuitry and adaptive equalization capability, the method comprising:	1, 3-5, receiver circuitry 30.
programming the programmable circuitry using configuration data with a first value corresponding to a first sampling location, wherein the programmable circuitry outputs the first value and a control signal;	*See, e.g.*, page 13, lines 6 to 27, and FIG. 5, programmable elements 186, element 184.
computing a second value corresponding to a second sampling location;	*See, e.g.*, page 13, lines 1 to 5, page 13, lines 6 to 27, and FIG. 5, element 188.
receiving the control signal, the first value and the second value in parallel;	*See, e.g.*, page 13, lines 6 to 27, and FIG. 5, multiplexer circuitry 182.
selecting, at the time the programmable circuitry is being programmed by the configuration data, based on the received control signal, one of the received first and second values, wherein the selecting selects one of the received first and second values only once only once while equalization implementation circuitry operates on the data signal;	*See, e.g.*, page 13, lines 6 to 27, and FIG. 5, multiplexer circuitry 182.
providing the selected one of the first and second values to the equalization implementation circuitry; and	*See, e.g.*, page 13, lines 6 to 27, and FIG. 5, multiplexer circuitry 182.
operating the equalization implementation circuitry, in the receiver circuitry,	*See, e.g.*, page 13, lines 6 to 27, and FIG. 5, multiplexer circuitry 182.

Claim 32	The Specification
using the sampling location corresponding to the selected one of the first and second values.	

(vi) GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The grounds of rejection to be reviewed on this appeal are: 1, 8, 12, 22, 24, 26, 27, 28, 30 and 32

1. The rejection of claims 1, 3-7, 26, 33, 38 and 41 under 35 U.S.C. § 103(a) as being allegedly obvious from Gorecki in view of WinSLAC and Solution Brief 41.

2. The rejection of claims 21, 22, 30 and 36 under 35 U.S.C. § 103(a) as allegedly being obvious from Jaynes in view of WinSLAC and Solution Brief 41.

3. The rejection of claims 20, 40 and 44 under 35 U.S.C. § 103(a) as allegedly being obvious from Jaynes in view of WinSLAC, Solution Brief 41 and Hillery.

4. The rejection of claims 8, 10, 11, 27, 34, 39 and 42 under 35 U.S.C. § 103(a) as allegedly being obvious from Gorecki and Lu in view of WinSLAC and Solution Brief 41.

5. The rejection of claims 24, 32, and 37 under 35 U.S.C. § 103(a) as allegedly being obvious from Gorecki and Lu in view of WinSLAC and Solution Brief 41.

6. The rejection of claims 12, 14-16, 28, 35, and 43 under 35 U.S.C. § 103(a) as allegedly being obvious from Pedersen in view of WinSLAC and Solution Brief 41.

(vii) ARGUMENT

A. Claims 1, 3-7, 26, 33, 38 and 41

Claims 1, 3-7, 26, 33, 38 and 41 have been rejected under 35 U.S.C. § 103(a) as being obvious from Gorecki in view

of WinSLAC and Solution Brief 41. This rejection is respectfully traversed.

Appellants are arguing each of dependent claims 3-7, 33, 38, and 41 together with independent claims 1 and 26. Appellants are not separately arguing the patentability of any dependent claim.

Appellants' invention, as defined by claims 1 and 26, is directed to receiver circuitry and methods for adaptively equalizing a data signal. The receiver circuitry and methods include, *inter alia*, programmable circuitry that is programmed with a first value corresponding to a first number of taps. Processing circuitry computes a second value corresponding to a second number of taps. Selection circuitry selects one of the first and second values at the time the programmable circuitry is being programmed by configuration data.

WinSLAC discusses a software tool that enables a user to design and generate an optimum set of coefficients for programmable filters that are modeled within the software (WinSLAC, page 1-2). Coefficients may be automatically calculated for a modeled filter or coefficient values may be manually entered by the software tool user (WinSLAC, pages 4-17, 4-18, and 4-19).

Solution Brief 41 discusses a FIR compiler wizard (software) that allows a user to create FIR filters. The software displays one selectable option to have filter coefficients provided from a file and a second option to have the coefficients generated by the compiler. The response of the filter can be viewed dynamically as the coefficient settings are changed. The wizard outputs code that can be

used to synthesize FIR filters of hardware architectures. (Solution Brief 41, FIG. 2 and pages 1 and 2.)

The Examiner acknowledges that Gorecki fails to show or suggest selection circuitry that selects based on a control signal one of a first value that is programmed and second value that is computed and cites WinSLAC and Solution Brief 41 as allegedly making up for this deficiency (Office Action, pages 3, 4, and 18). Appellants respectfully submit that Gorecki, WinSLAC and Solution Brief 41, alone or in combination, do not show or suggest selection circuitry that selects a first value that is programmed and a second value that is computed at the time the programmable circuitry is being programmed by configuration data, as defined by appellants' claims 1 and 26.

First, appellants respectfully submit that because WinSLAC and Solution Brief 41 each disclose a software tool that can be used to model various filter behavior and generate files to create a particular filter having the desired taps and coefficients, either of WinSLAC or Solution Brief 41 fails to show or suggest an already implemented receiver circuitry with selection circuitry. More specifically, the mere simulation and generation of files of receiver circuitry, as discussed in either WinSLAC or Solution Brief 41, is not the same as a circuit that is implemented to perform a particular function (e.g., select between two values), as defined by appellants' claims.

In particular, WinSLAC allows a user to select, using a user interface, to manually enter coefficients for a filter model or to have coefficients of the filter be automatically calculated by the software. Solution Brief 41 allows the user to select whether coefficient values are

provided by a file (specified by a user) or generated by the compiler (software) in order to simulate and synthesize (implement) a filter with a desired behavior. However, in either case, once the filter with the desired behavior is synthesized (implemented), the filter does not select between user provided coefficient values and software generated coefficient values as this selection is only performed during the simulation of the filter. Indeed, the SLAC menu software of WinSLAC and the compiler wizard of Solution Brief 41 (which each allow such selection and generates the alleged computed values) are not implemented in the filter and thus cannot provide the alleged computed coefficient values to the alleged selection circuitry in the implemented filter.

Second, WinSLAC discusses the user selecting between having coefficients automatically calculated for the filter (i.e., the alleged first value) and manually entering the filter coefficients (i.e., the alleged second value) by selecting respective options from a display screen displaying the SLAC menu. Solution Brief 41 discusses the user selecting whether the coefficient values are read from a file (i.e., the alleged first value) or generated using the compiler (i.e., the alleged second value) by selecting respective options from a display. Therefore, WinSLAC nor Solution Brief 41 necessarily does not show or suggest that the selection, between the first and second values, that is made by the circuitry is based on a control signal.

Finally, even if the user were analogized with selection circuitry that selects based on a control signal, each of WinSLAC and Solution Brief 41 would each still fail to show or suggest selection circuitry that selects one of two values at the time the programmable circuitry is being

programmed by configuration data, as required by appellants' claims 1 and 26. In particular, in each of WinSLAC and Solution Brief 41 the selection between the alleged two values is made while simulating the filter and before a configuration file of the filter is created. Therefore, because the alleged selection is made prior to the existence of the configuration data (i.e., before the compiler outputs the simulation file), the alleged selection is necessarily not made at the time the programmable circuitry is being programmed by configuration data.

Appellants remind the Examiner "[t]hat mere fact that a worker in the art could rearrange the parts of the reference device to meet the terms of the claims ... is not by itself sufficient to support a finding of obviousness. The prior art must provide a ... reason for the worker in the art, without the benefit of appellant's specification, to make the necessary changes in the reference device." Ex parte Chicago Rawhide Mfg. Co., 223 USPQ 351, 353; MPEP § 2144.04. The Examiner's position appears to require one to rearrange and modify the SLAC Menu software tool of WinSLAC and the displayed FIR compiler wizard of Solution Brief 41 to allow for the selection of one of a first value that is programmed and a second value that is computed (or one of a first and second training patterns) using physical selection circuitries allegedly shown or suggested by the other cited prior art. This modification is not supported by at least the WinSLAC and Solution Brief 41 references. On the contrary, the modification of SLAC Menu of WinSLAC or the FIR complier wizard of Solution Brief 41 would contravene the purpose of these cited references and change their principle operation at least because 1) such selection circuitry would be extraneous

to the operation of the software and 2) even if such selection circuitry were used to choose between two alleged filter coefficient values (i.e. values provided by a user or generated by software), it would obviate the user's ability to select between values which would render the software of WinSLAC or of Solution Brief 41 unusable. "If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims prima facie obvious." In re Ratti, 270 F.2d 810, 123 USPQ 349 (CCPA 1959). Thus, because combining a selection circuitry in the manner suggested by the Examiner would contravene the purpose of these cited references and change their principle operation, there is no reason for the worker in the art to make such changes without the benefit of appellants' specification. Therefore, the Examiner's position is insufficient as a matter of law to support a finding of obviousness.

Thus, either of WinSLAC and Solution Brief 41 do not make up for the deficiencies of Gorecki relative to the rejection. Therefore, Gorecki, WinSLAC, and Solution Brief 41 whether taken alone or in combination, do not show or suggest all the features of appellants' claims 1 and 26. Accordingly, appellants respectfully submit that independent claims 1 and 26 and claims 3-7, 33, 38 and 41 that depend, directly or indirectly from claim 1 or 26, are allowable.

B. Claims 8, 10, 11, 27, 34, 39 and 42

Claims 8, 10, 11, 27, 34, 39 and 42 have been rejected under 35 U.S.C. § 103(a) as being obvious from

Gorecki and Lu in view of WinSLAC and Solution Brief 41. This rejection is respectfully traversed.

Appellants are arguing each of dependent claims 10, 11, 34, 39, and 42 together with independent claims 8 and 27. Appellants are not separately arguing the patentability of any dependent claim.

Appellants' invention, as defined by independent claims 8 and 27, is directed to receiver circuitry and methods for adaptively equalizing a data signal. The receiver circuitry and methods include, *inter alia*, programmable circuitry that is programmed with a first value indicating a first selection between integer spacing and fractional spacing of the taps. Processing circuitry computes a second value indicating a first selection between integer spacing and fractional spacing of the taps. Selection circuitry selects one of the first and second values at the time the programmable circuitry is being programmed by configuration data.

The Examiner acknowledges that Gorecki fails to show or suggest selection circuitry that selects one of the first and second values based on a control signal and relies on WinSLAC and Solution Brief 41 to make up for this deficiency (Office Action, pages 11-13, 18, and 19).

As discussed above with respect to claims 1 and 26, appellants respectfully submit that WinSLAC and Solution Brief 41, alone or in combination, do not show or suggest selection circuitry that selects one of two values based on a control signal at the time the programmable circuitry is being programmed by configuration data. Thus, WinSLAC and Solution Brief 41, alone or in combination, do not

show or suggest these features defined by appellants' claims 8 and 27.

Lu does not make up for the deficiencies of Gorecki, WinSLAC and Solution Brief 41 relative to the rejection.

Therefore, Gorecki, WinSLAC, Solution Brief 41 and Lu, whether taken alone or in combination, do not show or suggest all the features of appellants' claims 8 and 27. Accordingly, appellants respectfully submit that independent claims 8 and 27 and claims 10, 11, 34, 39 and 42 that depend, directly or indirectly from claim 8 or 27, are allowable.

C. Claims 12, 14-16, 28, 35 and 43

Claims 12, 14-16, 28, 35 and 43 have been rejected under 35 U.S.C. § 103(a) as being obvious from Pedersen in view of WinSLAC and Solution Brief 41. Claims 16 and 43 have been rejected under 35 U.S.C. § 103(a) as being obvious from Pedersen in view of WinSLAC, Solution Brief 41 and Gorecki. These rejections are respectfully traversed.

Appellants are arguing each of dependent claims 14-16, 35, and 43 together with independent claims 12 and 28. Appellants are not separately arguing the patentability of any dependent claim.

Appellants' invention, as defined by independent claims 12 and 28, is directed to receiver circuitry and methods for adaptively equalizing a data signal. The receiver circuitry and methods include, inter alia, programmable circuitry that is programmed with a first starting value and outputs the first starting value and a control signal. Processing circuitry computes a second starting value and outputs the second starting value in parallel with the first starting value. Selection circuitry receives the control

signal and the first and second starting values in parallel and selects one of the first and second starting values based on the control signal at the time the programmable circuitry is being programmed by configuration data.

Appellants respectfully submit that Pedersen does not show or suggest programmable circuitry that is programmed by configuration data, as defined by appellants' claims 12 and 28. In particular, the Examiner alleges that a bank of settings available to the user in Pedersen is the same as appellants' claimed programmable circuitry (Office Action, page 14). However, contrary to the Examiner's allegations, nowhere does Pedersen show or suggest that the bank of settings is programmed by configuration data. Moreover, Pedersen discloses the bank of settings being made available to the user (and perhaps modified by the user) which is not the same as programming the bank of settings with configuration data. WinSLAC and Solution Brief 41 were cited as showing other features of appellants' claims and do not make up for the deficiencies of Pedersen in that regard.

In addition, the Examiner acknowledges that Pedersen fails to show or suggest selection circuitry that receives the first and second training patterns in parallel and selects one of the first and second values based on a control signal and relies on WinSLAC and Solution Brief 41 to make up for this deficiency (Office Action, pages 15 and 20).

As discussed above with respect to claims 1, 8, 26 and 27, WinSLAC and Solution Brief 41, alone or in combination, do not show or suggest selection circuitry that selects one of two values based on a control signal. Thus, WinSLAC and Solution Brief 41, alone or in combination, do not

show or suggest these features defined by appellants' claims 12 and 28.

Therefore, Pedersen, WinSLAC and Solution Brief 41, whether taken alone or in combination, do not show or suggest all the features of appellants' claims 12 and 28. Accordingly, appellants respectfully submit that claims 12 and 28, and claims 14-16, 35 and 43 that depend, directly or indirectly, from claim 12 or 28, are allowable.

D. Claims 21, 22, 30, and 36

Claims 21, 22, 30 and 36 have been rejected under 35 U.S.C. § 103(a) as being obvious from Jaynes in view of WinSLAC and Solution Brief 41.

Appellants are arguing each of dependent claims 21 and 36 together with independent claims 22 and 30. Appellants are not separately arguing the patentability of any dependent claim.

Appellants' invention, as defined by claims 22 and 30, is directed to receiver circuitry and a method for adaptively equalizing a data signal. The receiver circuitry and methods include, *inter alia*, programmable circuitry that is programmed with a first training pattern and outputs the first training pattern and a first control signal. Training pattern circuitry stores a second training pattern and outputs the second training pattern in parallel with the first training pattern. Selection circuitry receives the first control signal and the first and second training patterns in parallel and selects one of the first and second training pattern based on the first control signal at the time the programmable circuitry is being programmed by configuration data.

The Examiner acknowledges that Jaynes fails to show or suggest selection circuitry that receives the first and second training patterns in parallel and selects one of the first and second values based on a control signal and relies on WinSLAC and Solution Brief 41 to make up for this deficiency (Office Action, pages 6, 7, 20, and 21).

As discussed above with respect to claims 1, 12, 26 and 28 WinSLAC and Solution Brief 41, alone or in combination, do not show or suggest selection circuitry that selects one of two values based on a control signal. Thus, WinSLAC and Solution Brief 41, alone or in combination do not show or suggest these features defined by appellants' claims 22 and 30.

Therefore, Jaynes, WinSLAC and Solution Brief 41, whether taken alone or in combination, do not show or suggest all the features of appellants' claims 22 and 30. Accordingly, appellants respectfully submit that claims 22 and 30, and claims 21 and 36 that depend, directly or indirectly, from claims 22 or 30, are allowable.

E. Claims 24, 32 and 37

Claims 24, 32 and 37 were rejected under 35 U.S.C. § 103(a) as being obvious from Gorecki and Lu in view of WinSLAC and Solution Brief 41. This rejection is respectfully traversed.

Appellants are arguing dependent claim 37 together with independent claims 24 and 32. Appellants are not separately arguing the patentability of any dependent claim.

Appellants' invention, as defined by independent claims 24 and 32, is directed to receiver circuitry and methods for adaptively equalizing a data signal. The receiver

circuitry and methods include, *inter alia*, programmable circuitry that is programmed with a first value corresponding to a first sampling point location and outputs the first value and a control signal. Processing circuitry computes a second value corresponding to a second sampling point location and outputs the second starting value in parallel with the first starting value. Selection circuitry receives the control signal and the first and second values in parallel and selects one of the first and second values based on the control signal at the time the programmable circuitry is being programmed by configuration data.

The Examiner acknowledges that Gorecki fails to show or suggest selection circuitry that selects one of the first and second values based on a control signal and relies on WinSLAC and Solution Brief 41 to make up for this deficiency (Office Action, pages 10-14, and 21).

As discussed above with respect to claims 1, 8, 26 and 27 WinSLAC and Solution Brief 41, alone or in combination, do not show or suggest selection circuitry that selects one of two values based on a control signal. Thus, WinSLAC and Solution Brief 41, alone or in combination, do not show or suggest these features defined by appellants' claims 24, 32 and 37. Lu was cited by the Examiner as allegedly showing other features of appellants' claims and does not make up for the deficiencies of Gorecki, Solution Brief 41 and WinSLAC relative to the rejection. Accordingly, appellants respectfully submit that claims 24, 32 and 37 are allowable.

F. Claims 20, 40 and 44

Claims 20, 40 and 44 were rejected under 35 U.S.C. § 103(a) as being obvious from Jaynes in view of WinSLAC,

Solution Brief 41 and Hillery. This rejection is respectfully traversed. Appellants have shown claims 22 and 30 to be allowable, above. Appellants respectfully submit that claims 20, 40, and 44 that depend, directly or indirectly, from claims 22 or 30, are allowable.

G. Conclusion

For at least the reasons set forth above, appellants respectfully submit that the pending claims of this application are patentable, and that the rejections in the Office Action are improper and should be reversed. Prompt consideration of this appeal and reversal of the outstanding rejections are respectfully requested.

Respectfully submitted,

/Gall C. Gotfried/

Gall C. Gotfried
Registration No. 58,333
Agent for Appellants
Ropes & Gray LLP
Customer No. 75563
1211 Avenue of the Americas
New York, NY 10036-8704
Tel.: (212) 596-9000
Fax : (212) 596-9090

(viii) CLAIMS APPENDIX

CLAIMS APPENDIX A
CLAIMS ON APPEAL

1. Receiver circuitry for adaptively equalizing a data signal, the receiver circuitry comprising:

equalization implementation circuitry that includes a selectable number of taps, wherein the equalization implementation circuitry operates on the data signal;

programmable circuitry that is programmed by configuration data with a first value corresponding to a first number of taps;

processing circuitry that computes a second value corresponding to a second number of taps; and

selection circuitry that selects, based on a control signal, one of the first and second values as the selectable number of taps at the time the programmable circuitry is being programmed by the configuration data, wherein the selection circuitry selects one of the first and second values only once—while the equalization implementation circuitry operates on the data signal; and

wherein the selectable number of taps of the equalization implementation circuitry corresponds to the selected one of the first and second values.

3. The circuitry defined in claim 1 wherein the processing circuitry performs an algorithm to compute the second number.

4. A digital processing system comprising:

processor circuitry;

a memory coupled to the processor circuitry; and

the receiver circuitry as defined in claim 1 coupled to the processor circuitry and the memory.

5. A printed circuit board on which is mounted receiver circuitry as defined in claim 1.

6. The printed circuit board defined in claim 5 further comprising:

a memory mounted on the printed circuit board and coupled to the receiver circuitry.

7. The printed circuit board defined in claim 5 further comprising:

processor circuitry mounted on the printed circuit board and coupled to the receiver circuitry.

8. Receiver circuitry for adaptively equalizing a data signal, the receiver circuitry comprising:

equalization implementation circuitry that includes a filter with taps having a selected one of integer tap spacing and fractional tap spacing relative to the symbol rate of the data signal, wherein the equalization implementation circuitry operates on the data signal;

programmable circuitry that is programmed by configuration data with a first value indicating a first selection between integer spacing and fractional spacing of the taps;

processing circuitry that computes a second value indicating a second selection between integer spacing and fractional spacing of the taps; and

selection circuitry that selects one of the first and second values as the selected one of integer spacing and fractional spacing at the time the programmable circuitry is being programmed by the configuration data, wherein the selection circuitry selects one of the first and second values only once while the equalization implementation circuitry operates on the data signal; and

wherein the selection between the integer and fractional tap spacing corresponds to the selected one of the first and second values.

10. The circuitry defined in claim 8 wherein the processing circuitry performs an algorithm to compute the second selection.

11. The circuitry defined in claim 8 wherein the fractional spacing is a selectable fraction of the symbol period, wherein the first selection can include a programmably specified first fraction, and wherein the second selection can include a processing-circuitry-computed second fraction.

12. Receiver circuitry for adaptively equalizing a data signal, the receiver circuitry comprising:

equalization implementation circuitry that includes at least one selectable coefficient value;

first processing circuitry for computing the coefficient value using a selectable starting value, wherein the coefficient value is different from the starting value;

programmable circuitry that is programmed by configuration data with a first starting value and outputs the first starting value and a control signal;

second processing circuitry that computes a second starting value and outputs the second starting value in parallel with the first value; and

selection circuitry that:

receives the control signal from the programmable circuitry, the first starting value and the second starting value in parallel;

selects, based on the control signal, one of the first and second starting values at the time the programmable circuitry is being programmed by the configuration data, wherein the selection circuitry selects one of the first and second starting values only once; and

outputs the selected one of the first and second starting values to the first processing circuitry,

wherein the selectable starting value of the first processing circuitry corresponds to the selected one of the first and second values.

14. The circuitry defined in claim 12 wherein the first processing circuitry performs an algorithm to compute the coefficient value.

15. The circuitry defined in claim 12 wherein the second processing circuitry performs an algorithm to compute the second starting value.

16. The circuitry defined in claim 12 further comprising:

further programmable circuitry for allowing selection between (1) operation of the first processing circuitry to fix on the coefficient value that produces satisfactory equalization, and (2) continued operation of the first processing circuitry to continue to possibly adapt the coefficient value even after satisfactory equalization has been produced.

20. The circuitry defined in claim 40 wherein the second processing circuitry performs an algorithm to compute the second decision directed error signal.

21. The circuitry defined in claim 22 wherein the first processing circuitry performs an algorithm to compute the first error signal using a training pattern.

22. Receiver circuitry for adaptively equalizing a received data signal, the receiver circuitry comprising:

first processing circuitry for computing a first error signal using a selectable training pattern, wherein the first processing circuitry operates on the data signal;

programmable circuitry that is programmed by configuration data with a first training pattern and outputs the first training pattern and a first control signal;

training pattern circuitry that stores a second training pattern and outputs the second training pattern in parallel with the first training pattern; and

first selection circuitry that:

receives the first control signal from the programmable circuitry, the first training pattern and the second training pattern in parallel;

selects, based on the first control signal, one of the first and second training patterns at the time the programmable circuitry is being programmed by the

configuration data, wherein the first selection circuitry selects one of the first and second training patterns only once while the processing circuitry operates on the data signal; and

outputs the selected one of the first and second training patterns to the first processing circuitry.

24. Receiver circuitry for adaptively equalizing a data signal, the receiver circuitry comprising:

equalization implementation circuitry, in the receiver circuitry, having at least one sampling point with a selectable location relative to a bit period of the received signal, wherein the equalization implementation circuitry operates on the data signal;

programmable circuitry that is programmed by configuration data with a first value corresponding to a first location of the sampling point and outputs the first value and a control signal;

processing circuitry that computes a second value corresponding to a second location of the sampling point and outputs the second value in parallel with the first value; and

selection circuitry that:

receives the control signal from the programmable circuitry, the first value and the second value in parallel;

selects, based on the control signal, one of the first and second values at the time the programmable circuitry is being programmed by the configuration data, wherein the selection circuitry selects one of the first and second values only once, while the equalization implementation circuitry operates on the data signal; and

outputs the selected one of the first and second values to the equalization implementation circuitry,

wherein the location of the at least one sampling point of the equalization implementation circuitry corresponds to the selected one of the first and second values.

26. A method of operating receiver circuitry having programmable circuitry and adaptive equalization capability, the method comprising:

programming the programmable circuitry using configuration data with a first value corresponding to a first number of taps;

computing a second value corresponding to a second number of taps;

selecting, at the time the programmable circuitry is being programmed by the configuration data, one of the first and second values, wherein the selecting selects one of the first and second values only once while equalization implementation circuitry operates on a data signal;

providing the selected one of the first and second values to the equalization implementation circuitry; and

controlling the equalization implementation circuitry to operate with a number of taps corresponding to the selected one of the first and second values.

27. A method of operating receiver circuitry having programmable circuitry and equalization implementation circuitry that includes a filter with taps, the method comprising:

programming the programmable circuitry using configuration data with a first value indicating a first selection between integer spacing and fractional spacing of the taps;

computing a second value indicating a second selection between integer spacing and fractional spacing of the taps;

selecting, at the time the programmable circuitry is being programmed by the configuration data, based on the received control signal, one of the received first and second values, wherein the selecting selects one of the received first and second values only once while the equalization implementation circuitry operates on a data signal;

providing the selected one of the first and second values to the equalization implementation circuitry; and

controlling the filter of the equalization implementation circuitry, in the receiver circuitry, to operate with the tap spacing corresponding to the selected one of the first and second values.

28. A method of operating receiver circuitry having programmable circuitry and adaptive equalization capability, the method comprising:

programming the programmable circuitry using configuration data with a first starting value, wherein the

programmable circuitry outputs the first starting value and a control signal;

computing a second starting value;

receiving the control signal, the first starting value and the second starting value in parallel;

selecting, at the time the programmable circuitry is being programmed by the configuration data, based on the received control signal, one of the received first and second starting values, wherein the selecting selects one of the received first and second values only once while equalization implementation circuitry operates on a data signal;

processing the selected one of the first and second starting values to compute a coefficient value different from the selected starting value; and

operating the equalization implementation circuitry, in the receiver circuitry, using the computed coefficient.

30. A method of operating receiver circuitry having programmable circuitry and adaptive equalization capability, the method comprising:

programming the programmable circuitry using configuration data with a first training pattern, wherein the programmable circuitry outputs the first training pattern and a first control signal;

computing a second training pattern to output the second training pattern;

receiving the first control signal, the first starting training pattern and the second training pattern in parallel; and

selecting, based on the first control signal, one of the received first and second training patterns at the time the programmable circuitry is being programmed by the configuration data, wherein the selecting selects one of the first and second the training patterns only once, based on the configuration data, while equalization implementation circuitry operates on a data signal.

32. A method of operating receiver circuitry having programmable circuitry and adaptive equalization capability, the method comprising:

programming the programmable circuitry using configuration data with a first value corresponding to a

first sampling location, wherein the programmable circuitry outputs the first value and a control signal;

computing a second value corresponding to a second sampling location;

receiving the control signal, the first value and the second value in parallel;

selecting, at the time the programmable circuitry is being programmed by the configuration data, based on the received control signal, one of the received first and second values, wherein the selecting selects one of the received first and second values only once only once while equalization implementation circuitry operates on the data signal;

providing the selected one of the first and second values to the equalization implementation circuitry; and

operating the equalization implementation circuitry, in the receiver circuitry, using the sampling location corresponding to the selected one of the first and second values.

33. The circuitry defined in claim 1 wherein the programmable circuitry comprises first and second

programmable elements, wherein the first programmable element allows the first value to be specified and the second programmable element controls the selection made by the selection circuitry.

34. The circuitry defined in claim 8 wherein the programmable circuitry comprises first and second programmable elements, wherein the first programmable element allows the first value to be specified and the second programmable element controls the selection made by the selection circuitry.

35. The circuitry defined in claim 12 wherein the programmable circuitry comprises first and second programmable elements, wherein the first programmable element allows the first starting value to be specified and the second programmable element provides the first control signal.

36. The circuitry defined in claim 22 wherein the programmable circuitry comprises first and second programmable elements, wherein the first programmable element allows the first training pattern to be specified

and the second programmable element provides the first control signal.

37. The circuitry defined in claim 24 wherein the programmable circuitry comprises first and second programmable elements, wherein the first programmable element allows the value to be specified and the second programmable element provides the control signal.

38. The circuitry defined in claim 1 wherein:

the programmable circuitry outputs the first value and a control signal;

the processing circuitry outputs the second value in parallel with the first value;

the selection circuitry:

receives the control signal from the programmable circuitry, the first value and the second value in parallel;

selects, based on the received control signal, one of the first and second values; and

outputs the selected one of the first and second values to the equalization implementation circuitry.

39. The circuitry defined in claim 8 wherein:

the programmable circuitry outputs the first value and a control signal;

the processing circuitry outputs the second value in parallel with the first value; and

the selection circuitry:

receives the control signal from the programmable circuitry, the first value and the second value in parallel;

selects, based on the control signal, one of the first and second values; and

outputs the selected one of the first and second values to the equalization implementation circuitry.

40. The circuitry defined in claim 22 further comprising:

equalization implementation circuitry responsive to an error signal, wherein the equalization implementation circuitry operates on the data signal;

wherein the first processing circuitry receives the selected one of the first and second training patterns

and computes the first error signal using the selected training pattern and outputs the first error signal;

second processing circuitry that computes a second decision directed error signal using a training pattern and outputs the second error signal in parallel with the first error signal; and

second selection circuitry that:

receives a second control signal from the programmable circuitry, the first error signal and the second error signal in parallel;

selects, based on the second control signal, one of the first and second error signals; and

outputs the selected one of the first and second error signals to the equalization implementation circuitry,

wherein the equalization implementation circuitry is responsive to the selected one of the first and second error signals.

41. The method of claim 26 wherein:

the programmable circuitry outputs the first value and a control signal; and

the control signal, the first value and the second value are received in parallel.

42. The method of claim 27 wherein:

the programmable circuitry outputs the first value and a control signal; and

the control signal, the first value and the second value are received in parallel.

43. The method of claim 28 further comprising:

selecting, at the time the programmable circuitry is being programmed by the configuration data, whether the computed coefficient to be used in equalization implementation circuitry is to be determined once or on an on-going basis; and

determining the computed coefficient in accordance with the selecting.

44. The method of claim 30 further comprising:

computing a first error signal based on the selected training pattern;

computing a second decision directed error signal based on a training pattern;

receiving the second control signal, the first error signal and the second error signal in parallel;

selecting, based on a second control signal provided by the programmable circuitry, one of the received first and second error signals; and

using the selected one of the first and second error signals in a determination of at least one operating parameter of the equalization implementation circuitry.

(ix) EVIDENCE APPENDIX

APPENDIX B

COPY OF THE FINAL OFFICE ACTION DATED OCTOBER 15, 2009

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/762,864	01/21/2004	Wilson Wong	174/295	5928

36981 7590 10/15/2009

ROPES & GRAY LLP
PATENT DOCKETING 39/361
1211 AVENUE OF THE AMERICAS
NEW YORK, NY 10036-8704

EXAMINER	
FOTAKIS, ARISTOCRATIS	
ART UNIT	**PAPER NUMBER**
2611	

MAIL DATE	DELIVERY MODE
10/15/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/762,864	WONG ET AL.	
	Examiner	Art Unit	
	ARISTOCRATIS FOTAKIS	2611	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07/13/2009*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1, 3 - 8, 10 - 12, 14 - 16, 20 - 22, 24, 26 - 28, 30, 32 - 44* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1, 3 - 8, 10 - 12, 14 - 16, 20 - 22, 24, 26 - 28, 30, 32 - 44* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to

consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Claims 1, 3 – 7, 26, 33, 38 and 41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gorecki (US 20040071205) in view of WinSLAC Software User's Guide (1999) and further in view of Solution Brief 41 ("FIR Compiler MegaCore Function", Altera Corporation, June 1999, ver.1).

Re claims 1, 26, 33, 38 and 41, Gorecki teaches of a circuitry (transceiver, Fig.4) for adaptively equalizing a data signal, the circuitry (Abstract) comprising: equalization implementation circuitry that includes a selectable tap parameter (*positioning of taps, pulse duration of taps, tap coefficients*), wherein the equalization implementation circuitry operates on the data signal (Paragraph 0042, 0043); programmable circuitry that is programmed by configuration data with a first value corresponding to a first tap parameter (*user*, Paragraph 0046); processing circuitry that computes a second value corresponding to a second tap parameter (*adaptive algorithm*, Paragraph 0044 – 0045); the user or system may select between the first and second numbers as the selectable number of taps selecting one of the first and second values only once (*initialization or start-up*, Paragraph 0112). However, Gorecki does not specifically teach of the selection circuitry in a receiver circuitry based on a control signal selecting one of the first and second values and the number of taps being a tap parameter that is programmed or processed.

WinSLAC Software User's Guide discloses of equalization for both receive and transmit paths (page 4-15) controlled by user interface dialogs in the WinSLAC software (Page 3-5). The Guide further discloses of a selection circuitry based on a control signal (user interface) selecting one of a first value (*Calc or calculate*, 4.6 SLAC Menu, Pages 4 – 17 to 4 - 19) and a second value (*Set*, 4.6 SLAC Menu, Pages 4 – 17 to 4 - 19). However, WinSLAC Software User's Guide does not specifically show of the selection circuitry based on a control signal (user interface) selecting one of a first value that has been programmed and a second value that has been computed and that the number of taps being a tap parameter that is programmed or processed.

Solution Brief 41discloses of a FIR Compiler that identifies coefficients that match the frequency response specifies by the system. The coefficients can be read from a file or generated using the FIR compiler wizard. The function lets you specify the sample rate, the number of taps and cut-off frequencies. As you change the coefficient settings, you can view the frequency and the response of the filter dynamically (Pages 1 – 2, Fig.2).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to have the equalization circuitry on the receiver so as to be able to monitor or compensate fast varying channel conditions. It would have been obvious to one having ordinary skill in the art at the time the invention was made to have the selection circuitry select either the programmable circuitry or the processing circuitry controlled by the user to calculate or program tap parameters in order to provide a more flexible and user-defined system. It would have been obvious to one having ordinary

skill in the art at the time the invention was made to have had the two circuitries store the values into memory files before selection so as to avoid recalculations of the values. It would have been obvious to one having ordinary skill in the art at the time the invention was made to have programmed or calculated the number of taps as an essential tap coefficient setting since it is required to know the exact number of taps before finding the tap coefficients.

Re claim 3, Gorecki teaches of the processing circuitry performing an algorithm to compute the second number (Paragraph 0045).

Re claim 4, Gorecki teaches of a memory coupled to the processor programmable logic device circuitry coupled to the processor circuitry and the memory (Paragraph 0112).

Re claims 5 - 6, Gorecki teaches of a printed circuit board comprising: a memory mounted on the printed circuit board and coupled to the programmable logic device circuitry (Paragraph 0112).

Re claim 7, Gorecki teaches of the printed circuit board further comprising: processor circuitry mounted on the printed circuit board and coupled to the programmable logic device circuitry (Paragraph 0112).

Claims 21 – 22, 30 and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Jaynes et al (US US 2005/0047779) in view of WinSLAC Software User's Guide (1999) and further in view of Solution Brief 41 ("FIR Compiler MegaCore Function", Altera Corporation, June 1999, ver.1).

Re claims 22, 30 and 36, Jaynes teaches of a receiver circuitry ([0011]) for adaptively equalizing a data signal (Paragraph 0008, Figure) comprising: a first processing circuitry for computing an error signal using a selectable training pattern (#70, #72, Figure), wherein the first processing circuitry operates on the received data signal (Figure); programmable circuitry that is programmed by configuration data with a first training pattern and outputs the first training pattern (operator, Figure); training pattern circuitry that outputs a second training pattern (external process, Figure); and a first selection circuitry that receives the first training pattern and second training pattern; selects one of the first and second training patterns (external process or operator, Figure, Paragraph 0023) at the time the programmable circuitry is being programmed by configuration data (*user,* Paragraphs 0046, 0050, 0112), wherein the selection circuitry selects one of the first and second training patterns only once while the processing circuitry operates the data signal (Paragraphs 0008, 0022 – 0023, 0026) and outputs the selected one of the first and second training patterns to the first processing circuitry..

However, Jaynes does not specifically teach of the selection circuitry receiving the first training pattern and second training pattern in parallel and selecting one of the training patterns based on a control signal from the programmable circuitry.

WinSLAC Software User's Guide discloses of equalization for both receive and transmit paths (page 4-15) controlled by user interface dialogs in the WinSLAC software (Page 3-5). The Guide further discloses of a selection circuitry based on a control signal (user interface) selecting one of a first value (*Calc or calculate*, 4.6 SLAC Menu, Pages 4 – 17 to 4 - 19) and a second value (*Set*, 4.6 SLAC Menu, Pages 4 – 17 to 4 - 19). However, WinSLAC Software User's Guide does not specifically disclose of the two values are in parallel before selection.

Solution Brief 41 discloses of a FIR Compiler that identifies coefficients that match the frequency response specifies by the system. The coefficients can be read from a file or generated using the FIR compiler wizard. The function lets you specify the sample rate, the number of taps and cut-off frequencies. As you change the coefficient settings, you can view the frequency and the response of the filter dynamically (Pages 1 – 2, Fig.2).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to have the selection circuitry controlled by the user to select between a processed value and a programmers value in order to provide a more flexible and user-defined system. It would have been obvious to one having ordinary skill in the art at the time the invention was made to have had the two circuitries store the values into memory files before selection so as to avoid recalculations of the values.

Re claim 21, Jaynes teaches of the first processing circuitry performing an algorithm to compute the error signal using a training pattern (*the error generator is a processor*, Paragraph 0015).

Claims 20, 40 and 44 are rejected under 35 U.S.C. 103(a) as being unpatentable over Jaynes, WinSLAC Software User's Guide and Solution Brief 41 in view of Hillery (US 6,178,201).

Re claims 40 and 44, Jaynes, WinSLAC Software User's Guide and Solution Brief 41 disclose all the limitations of claims 22 and 30 as well as Jaynes teaching of equalization implementation circuitry (#30) responsive to an error signal (#70), wherein the equalization implementation circuitry operates on the received data signal. As discussed above, Jaynes, WinSLAC Software User's Guide and Solution Brief 41 teach of wherein a processing circuitry receives the selected one of the first and second training patterns and computes the first error signal using the selected training pattern and outputs the first error signal. Jaynes further teaches of the processing circuitry can also compute a second decision directed error signal using a training pattern (#81) and outputs the second error signal. However, Jaynes does not specifically disclose of a separate (second) processing circuitry that computes a second decision directed error signal using a training pattern and outputs the second error signal in parallel with the first error signal; and second selection circuitry that: receives a second control signal from the programmable circuitry, the first error signal and the second error signal in

parallel; selects, based on the second control signal, one of the first and second error signals; and outputs the selected one of the first and second error signals to the equalization implementation circuitry, wherein the equalization implementation circuitry is responsive to the selected one of the first and second error signals.

Hillery teaches of a receiver circuitry for adaptively equalizing a data signal (Abstract) comprising: equalization implementation circuitry responsive to an error signal (Fig.1), wherein the equalization implementation circuitry operates on the received data signal (#22, Fig.1); first processing circuitry for computing a first decision directed error signal (#40, Fig.1, Col 3, Lines 54 – 67); second processing circuitry for computing a second error (#38, Fig.1, Col 3, Lines 38 – 50); and selection circuitry (#36, Fig.1) for selecting one of the first and second error signals as the error signal (Col 3, Lines 30 – 37), and outputs the selected one of the first and second error signals to the equalization implementation circuitry (#30), wherein the equalization implementation circuitry is responsive to the selected one of the first and second error signals (Fig.1). However, Hillery does not teach of the selector selecting based on a second control signal from the programmable circuitry.

WinSLAC Software User's Guide discloses of equalization for both receive and transmit paths (page 4-15) controlled by user interface dialogs in the WinSLAC software (Page 3-5). The Guide further discloses of a selection circuitry based on a control signal (user interface) selecting one of a first value (*Calc or calculate*, 4.6 SLAC Menu, Pages 4 – 17 to 4 - 19) and a second value (*Set*, 4.6 SLAC Menu, Pages 4 – 17 to 4 - 19).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to have had two processing circuitries to generate the error signals so as to speed up the adaptation of the equalizer. It would have been obvious to one having ordinary skill in the art at the time the invention was made to have the selection circuitry controlled by the user to select between a first value and a secondvalue in order to provide a more flexible and user-defined system.

Re claim 20, Hillery teaches of wherein the second processing circuitry performs an algorithm to compute the first second decision directed error signal (LMS, Col 3, Lines 50 - 65).

Claims 8 – 11, 24, 27, 32, 34, 37, 39 and 42 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gorecki in view of Lu (US 6,275,836) and further in view of WinSLAC Software User's Guide (1999) and further in view of Solution Brief 41.

Re claims 8, 10, 27, 34, 39 and 42, Gorecki teaches of a circuitry (transceiver, Fig.4) for adaptively equalizing a data signal comprising: equalization implementation circuitry for adjusting or controlling the spacing of the taps (Paragraph 0043), wherein the equalization implementation circuitry operates on the data signal; programmable circuitry for adjusting or controlling the spacing of the taps programmed by configuration data (*user,* Paragraphs 0046, 0050, 0112); processing circuitry for adjusting or controlling the spacing of the taps (Paragraph 0044 – 0045); the

programmable circuitry or the processing circuitry may generate one of the tap spacing (Paragraph 0112) only once (*initialization or start-up,* Paragraph 0112), while the equalization implementation circuitry operates on the data signal (Paragraph 0112 – 0116). However, Gorecki does not teach of a programmable circuitry and processing circuitry for allowing a first selection between integer spacing and fractional spacing to be specified as well as the selection circuitry selecting either the output of the programmable circuitry or the processing circuitry but the equalization effects performed by either one of the two circuitries. However, Gorecki does not specifically teach of the selection circuitry in a receiver circuitry based on a control signal selecting one of the first and second values.

Lu teaches of a programmable logic device circuitry for adaptively equalizing a received data signal (Abstract, Fig.3) comprising: equalization implementation circuitry including taps (interpolation filter) having a selected one of integer spacing and fractional spacing relative to the symbol rate of the data signal (Abstract, Lines 1 – 13, Fig.3); processing circuitry (#74, Fig.3) for computing a (second) selection (#76a, #76b, Fig.3) between integer spacing and fractional spacing (Abstract, Lines 9 – 13, Fig.3 and Col 7, Lines 17 – 29).

WinSLAC Software User's Guide discloses of equalization for both receive and transmit paths (page 4-15) controlled by user interface dialogs in the WinSLAC software (Page 3-5). The Guide further discloses of a selection circuitry based on a control signal (user interface) selecting one of a first value (*Calc or calculate*, 4.6 SLAC Menu, Pages 4 – 17 to 4 - 19) and a second value (*Set*, 4.6 SLAC Menu, Pages 4 – 17 to 4 - 19).

However, WinSLAC Software User's Guide does not specifically show of the selection circuitry based on a control signal (user interface) selecting one of a first value that has been programmed and a second value that has been computed.

Solution Brief 41 discloses of a FIR Compiler that identifies coefficients that match the frequency response specifies by the system. The coefficients can be read from a file or generated using the FIR compiler wizard. The function lets you specify the sample rate, the number of taps and cut-off frequencies. As you change the coefficient settings, you can view the frequency and the response of the filter dynamically (Pages 1 – 2, Fig.2).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to have provided the option to the user to choose between a fixed or fractional spacing depending on the incoming sampling rate for a good equalizer performance. It would have been obvious to one having ordinary skill in the art at the time the invention was made to have the selection circuitry selecting either the output of the programmable circuitry or the processing circuitry controlled by the user in order to provide a more flexible and user-defined system. It would have been obvious to one having ordinary skill in the art at the time the invention was made to have had the two circuitries store the values into memory files before selection so as to avoid recalculations of the values. It would have been obvious to one having ordinary skill in the art at the time the invention was made to have the equalization circuitry on the receiver so as to be able to monitor or compensate fast varying channel conditions.

Re claim 11, Gorecki, Lu, WinSLAC Software User's Guide and Solution Brief 41 teach of all the limitations of claim 8. Lu teaches of the fractional spacing is a selectable fraction of the symbol period (1/fs, sampling rate fs, Col 7, Lines 48 – 62), wherein the first selection can include a programmably specified first fraction, and wherein the second selection can include a processing-circuitry-computed second fraction (see claim rejection above).

Re claims 24, 32 and 37 Gorecki, Lu, WinSLAC Software User's Guide and Solution Brief 41 teach of a receiver circuitry for adaptively equalizing a data signal as discussed above in claims 8 – 11, comprising: equalization implementation circuitry, in the receiver circuitry, having at least one sampling point with a selectable location relative to a bit period of the received signal (*The symbol period of the tap spacing's is the inverse of the sampling frequency. Changing the tap spacing (as taught by Gorecki) will change the location of the sampling points*), wherein the equalization implementation circuitry operates on the data signal; programmable circuitry that is programmed by configuration data with a first value corresponding to a first location of the sampling point and outputs the first value and a control signal; processing circuitry that computes a second value corresponding to a second location of the sampling point and outputs the second value in parallel with the first value; and selection circuitry that: receives the control signal from the programmable circuitry, the first value and the second value in parallel (*receiving two values in parallel and selecting only one as*

disclosed by WinSLAC Software User's Guide); selects, based on the control signal, one of the first and second values at the time the programmable circuitry is being programmed by the configuration data, wherein the selection circuitry selects one of the first and second values only once, while the equalization implementation circuitry operates on the data signal; and outputs the selected one of the first and second values to the equalization implementation circuitry, wherein the location of the at least one sampling point of the equalization implementation circuitry corresponds to the selected one of the first and second values.

Claims 12, 14 – 16, 28, 35 and 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Pederson et al (US 2006/0114979) in view of WinSLAC Software User's Guide (1999) and further in view of Solution Brief 41 ("FIR Compiler MegaCore Function", Altera Corporation, June 1999, ver.1).

Re claims 12, 28 and 35, Pederson teaches of circuitry for adaptively equalizing a data signal, the circuitry comprising: equalization implementation circuitry that includes at least one selectable coefficient value (FCS, Fig.6); first processing circuitry for computing the coefficient value (UPS) using a selectable starting value (iFCS), wherein the coefficient value (FCS) is different from the starting value (iFCS); programmable circuitry that is programmed by configuration data with a first starting value and outputs the first starting value (Paragraph 0104, Lines 6- 8); second processing circuitry that computes a second starting value and outputs the second

starting value (algorithm, Paragraph 0104, Lines 4 - 6); selects one of the first and second starting values (Paragraph 0104), only once (initial values); and outputs the selected one of the first and second starting values to the first processing circuitry (UPS), wherein the selectable starting value of the first processing circuitry corresponds to the selected one of the first and second values (Paragraph 0102 - 0105, Fig.6b). However, Pederson does not specifically of the programmable circuitry outputting a control signal used to select one of the two parallel values in a receiver.

WinSLAC Software User's Guide discloses of equalization for both receive and transmit paths (page 4-15) controlled by user interface dialogs in the WinSLAC software (Page 3-5). The Guide further discloses of a selection circuitry based on a control signal (user interface) selecting one of a first value (*Calc or calculate*, 4.6 SLAC Menu, Pages 4 – 17 to 4 - 19) and a second value (*Set*, 4.6 SLAC Menu, Pages 4 – 17 to 4 - 19). However, WinSLAC Software User's Guide does not specifically show of the selection circuitry based on a control signal (user interface) selecting one of a first value that has been programmed and a second value that has been computed in parallel.

Solution Brief 41discloses of a FIR Compiler that identifies coefficients that match the frequency response specifies by the system. The coefficients can be read from a file or generated using the FIR compiler wizard. The function lets you specify the sample rate, the number of taps and cut-off frequencies. As you change the coefficient settings, you can view the frequency and the response of the filter dynamically (Pages 1 – 2, Fig.2).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to have the equalization circuitry on the receiver so as to be able to monitor or compensate fast varying channel conditions. It would have been obvious to one having ordinary skill in the art at the time the invention was made to have the selection circuitry select either the programmable circuitry or the processing circuitry controlled by the user to calculate or program tap parameters in order to provide a more flexible and user-defined system. It would have been obvious to one having ordinary skill in the art at the time the invention was made to have had the two circuitries store the values into memory files before selection so as to avoid recalculations of the values.

Re claim 14, Pederson teaches of wherein the first processing circuitry performs an algorithm to compute the coefficient value (Abstract).

Re claim 15, Pederson teaches of wherein the second processing circuitry performs an algorithm to compute the second starting value (Paragraph 0104).

Claims 16 and 43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Pederson, WinSLAC Software User's Guide (1999) and Solution Brief 41 in view of Gorecki.

Pederson, WinSLAC Software User's Guide (1999) and Solution Brief 41 teach all the limitations of claims 12 and 28 except of further comprising: further programmable circuitry for allowing selection between (I) operation of the first

processing circuitry to fix on the coefficient value that produces satisfactory equalization, and (2) continued operation of the first processing circuitry to continue to possibly adapt the coefficient value even after satisfactory equalization has been produced.

Gorecki teaches of selection between (1) operation of the first processing circuitry to fix (*adjust*) on the coefficient value that produces satisfactory equalization, and (2) continued operation of the first processing circuitry to continue to possibly adapt (*control or vary*) the coefficient value even after satisfactory equalization has been produced (paragraphs 0047, 0058, 0062 – 0065 and 0070).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to have the user select between a fixed and continuous operation for the benefits of having a more flexible and user-defined system.

Response to Arguments

Applicant's arguments filed July 13, 2009 have been fully considered but they are not persuasive.

Re claims 1, 8, 26 and 27, Applicants submit that Solution Brief 41 fails to show or suggest an already implemented receiver circuitry with selection circuitry. Solution Brief 41 would still fail to show or suggest selection circuitry that selects one of two

values at the time the programmable circuitry is being programmed by configuration data.

In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). Gorecki and WinSLAC show or suggest an already implemented receiver circuitry with selection circuitry as well as showing or suggesting selection circuitry that selects one of two values at the time the programmable circuitry is being programmed by configuration data. Solution Brief 41 shows that filter parameters can be read from a file or generated using the compiler wizard. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to have had the two circuitries of Gorecki and WinSLAC to store the values into memory files before selection so as to avoid recalculations of the values.

Re claims 1, 8, 26 and 27, Applicants submit that Solution Brief 41 necessarily does not show or suggest that the selection, between the first and second values, that is made by the circuitry is based on a control signal.

Examiner submits that both WinSLAC and Solutions Brief 41 teach of the selection, between the first and second values, that is made by the circuitry is based on a control signal. WinSLAC discloses of a selection circuitry based on a control signal (user interface) selecting one of a first value (*Calc or calculate*, 4.6 SLAC Menu, Pages

4 – 17 to 4 - 19) and a second value (*Set*, 4.6 SLAC Menu, Pages 4 – 17 to 4 - 19). Solutions Brief 41 shows a MegaWizard Plug-In Manager-FIR Compiler window where the selection can be achieved by the user with the press of a button. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

Re claims 12 and 28, Applicants submit that nowhere does Pederson show or suggest that the bank of settings is programmed by configuration data. Moreover, Pederson discloses the bank of settings being made available to the user (and perhaps modified by the user) which is not the same as programming the bank of settings with configuration data.

Examiner submits that Pederson teaches of the initial settings may be established on the basis of complex filter design algorithms **or** they may represent for example settings of preferred filters, earlier tested and approved by the user (Paragraph 0104). Therefore, Pederson discloses the bank of settings being made available to the user and programmed with configuration data by the user.

Re claims 12 and 28, Applicants submit that Solution Brief 41 necessarily does not show or suggest that the selection, between the first and second values, that is made by the circuitry is based on a control signal.

Examiner submits that both WinSLAC and Solutions Brief 41 teach of the selection, between the first and second values, that is made by the circuitry is based on a control signal. WinSLAC discloses of a selection circuitry based on a control signal (user interface) selecting one of a first value (*Calc or calculate*, 4.6 SLAC Menu, Pages 4 – 17 to 4 - 19) and a second value (*Set*, 4.6 SLAC Menu, Pages 4 – 17 to 4 - 19). Solutions Brief 41 shows a MegaWizard Plug-In Manager-FIR Compiler window where the selection can be achieved by the user with the press of a button. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

Re claims 22 and 30, Applicants submit that Solution Brief 41 necessarily does not show or suggest that the selection, between the first and second values, that is made by the circuitry is based on a control signal.

Examiner submits that both WinSLAC and Solutions Brief 41 teach of the selection, between the first and second values, that is made by the circuitry is based on a control signal. WinSLAC discloses of a selection circuitry based on a control signal (user interface) selecting one of a first value (*Calc or calculate*, 4.6 SLAC Menu, Pages

4 – 17 to 4 - 19) and a second value (*Set*, 4.6 SLAC Menu, Pages 4 – 17 to 4 - 19). Solutions Brief 41 shows a MegaWizard Plug-In Manager-FIR Compiler window where the selection can be achieved by the user with the press of a button. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

Re claims 24 and 32, Applicants submit that Solution Brief 41 necessarily does not show or suggest that the selection, between the first and second values, that is made by the circuitry is based on a control signal.

Examiner submits that both WinSLAC and Solutions Brief 41 teach of the selection, between the first and second values, that is made by the circuitry is based on a control signal. WinSLAC discloses of a selection circuitry based on a control signal (user interface) selecting one of a first value (*Calc or calculate*, 4.6 SLAC Menu, Pages 4 – 17 to 4 - 19) and a second value (*Set*, 4.6 SLAC Menu, Pages 4 – 17 to 4 - 19). Solutions Brief 41 shows a MegaWizard Plug-In Manager-FIR Compiler window where the selection can be achieved by the user with the press of a button. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aristocratis Fotakis whose telephone number is (571) 270-1206. The examiner can normally be reached on Monday - Thursday 6:30 - 4.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Chieh M. Fan can be reached on (571) 272-3042. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Aristocratis Fotakis/

Examiner, Art Unit 2611

/CHIEH M FAN/

Supervisory Patent Examiner, Art Unit 2611

APPENDIX C
COPY OF U.S. PATENT APPLICATION
PUBLICATION NO. 2004/0071205 ("GORECKI")

US 20040071205A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2004/0071205 A1

Gorecki (43) Pub. Date: Apr. 15, 2004

(54) SYSTEM AND METHOD OF EQUALIZATION OF HIGH SPEED SIGNALS

(76) Inventor: James Gorecki, Hillsboro, OR (US)

Correspondence Address:
NEIL STEINBERG
STEINBERG & WHITT, LLP
2665 MARINE WAY, SUITE 1150
MOUNTAIN VIEW, CA 94043 (US)

(21) Appl. No.: 10/412,101

(22) Filed: Apr. 10, 2003

Related U.S. Application Data

(62) Division of application No. 10/269,446, filed on Oct. 11, 2002.

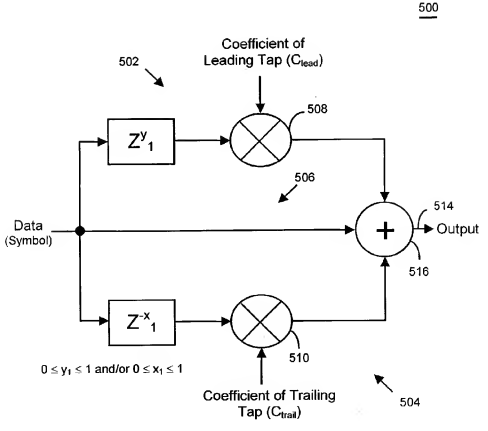
Publication Classification

(51) Int. Cl.[7] H03H 7/30; H03K 5/159; H03H 7/40; H04L 27/20; H03K 9/02; H03K 7/02

(52) U.S. Cl. 375/232; 375/295; 375/353

(57) ABSTRACT

In one aspect, the present invention is directed to a technique of, and system for enhancing the performance of high-speed digital communications through a communications channel, for example a backplane. In this aspect of the present invention, a transmitter includes equalization circuitry to compensate for bandwidth limitations and reflections in high-speed digital communication systems. In one embodiment, the equalization circuitry is designed, programmed and/or configured to introduce intersymbol interference in order to improve the signal integrity in high-speed communications and enhance the operation and performance of such systems. In this regard, the equalization circuitry includes temporally overlapping leading and/or trailing taps (relative to the data (symbol) signal) to reduce, minimize, mitigate or effectively eliminate pre-cursor and/or post-cursor intersymbol interference due to, for example, bandwidth limitations and reflections in high-speed digital communication systems. The amount of equalization may be programmed, adjusted or controlled by varying the positioning of the tap(s), varying the coefficients of the tap(s), and/or varying the pulse durations of the tap(s) (that is, the pulse duration of the equalization signal attributed to the tap).



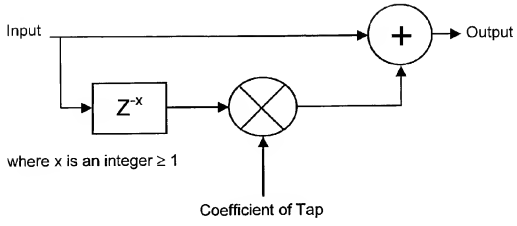


FIGURE 1

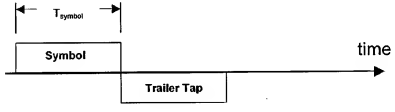


FIGURE 2

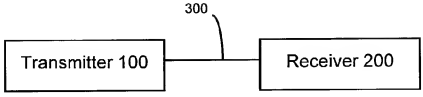


FIGURE 3

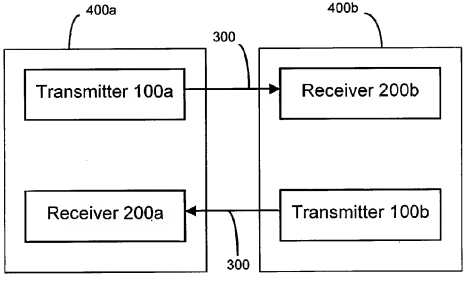


FIGURE 4

FIGURE 5A

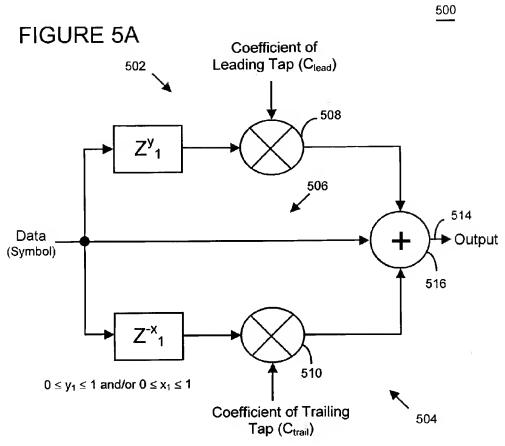


FIGURE 6

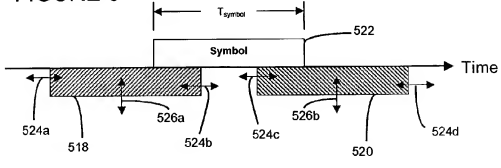
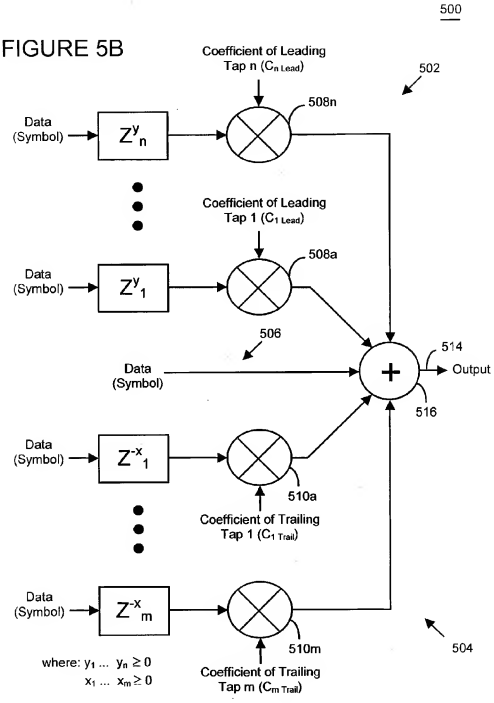


FIGURE 5B


500
Coefficient of Leading Tap n ($C_{n\ Lead}$)
502
508n
Data (Symbol)
Z^{y}_{n}
Coefficient of Leading Tap 1 ($C_{1\ Lead}$)
508a
Data (Symbol)
Z^{y}_{1}
506
514
Data (Symbol)
Output
516
Data (Symbol)
Z^{-x}_{1}
510a
Coefficient of Trailing Tap 1 ($C_{1\ Trail}$)
Data (Symbol)
Z^{-x}_{m}
510m
504
where: $y_1 \ldots y_n \geq 0$
$x_1 \ldots x_m \geq 0$
Coefficient of Trailing Tap m ($C_{m\ Trail}$)

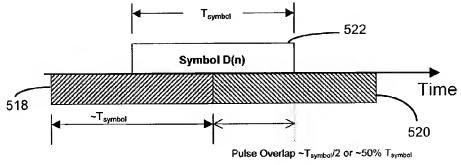


FIGURE 7

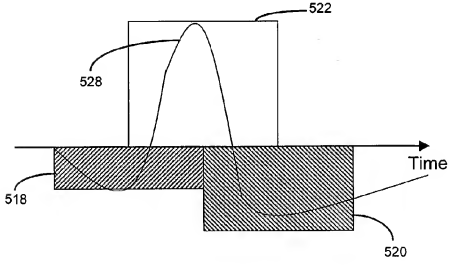


FIGURE 8

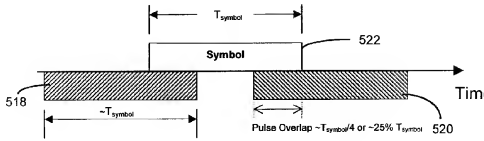


FIGURE 9A

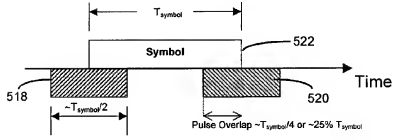


FIGURE 9B

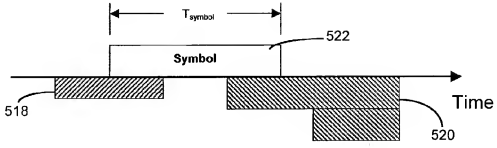


FIGURE 9C

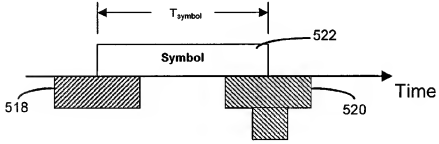


FIGURE 9D

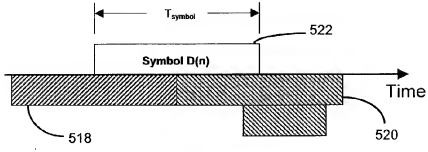


FIGURE 9E

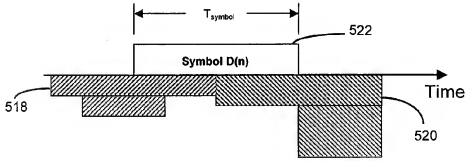


FIGURE 9F

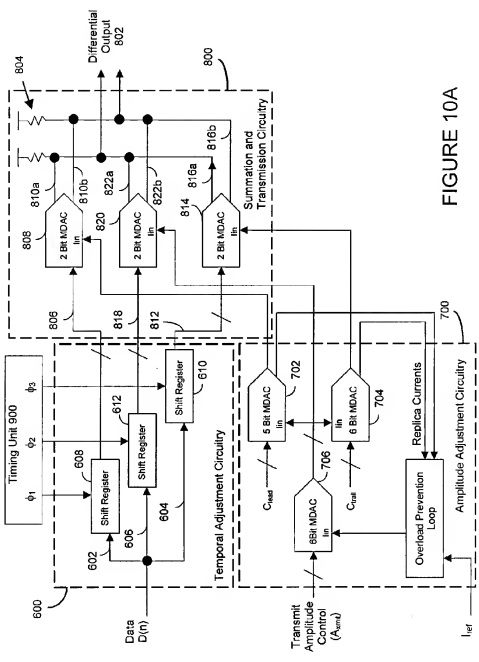

Differential Output 802
804
800
Summation and Transmission Circuitry
2 Bit MDAC
808
810a
810b
820
822a
822b
814
816a
816b
806
818
812
700
Timing Unit 900
ϕ_1
ϕ_2
ϕ_3
Shift Register
608
612
610
602
606
604
Temporal Adjustment Circuitry
5 Bit MDAC
702
6 Bit MDAC
704
8Bit MDAC
706
C_{mat}
Replica Currents
Overload Prevention Loop
Amplitude Adjustment Circuitry
600
Data D(n)
Transmit Amplitude Control (A_{amp})
I_{ref}

FIGURE 10A

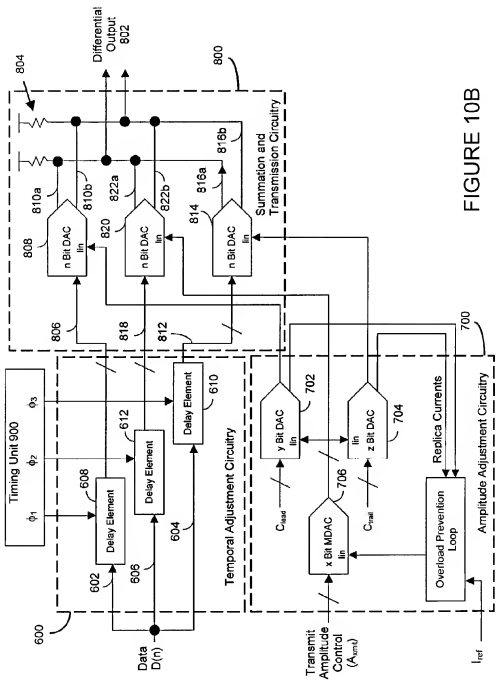

Differential Output 802
804
800
Summation and Transmission Circuitry
n Bit DAC
Iin
808
810a
810b
820
822a
822b
814
816a
816b
806
818
812
Timing Unit 900
ϕ_1
ϕ_2
ϕ_3
608
612
610
Delay Element
602
606
604
Temporal Adjustment Circuitry
600
Data D(n)
y Bit DAC
702
z Bit DAC
704
x Bit MDAC
706
C_{bias}
C_{mult}
Transmit Amplitude Control (A_{cntrl})
700
Replica Currents
Overload Prevention Loop
Amplitude Adjustment Circuitry
I_{ref}

FIGURE 10B

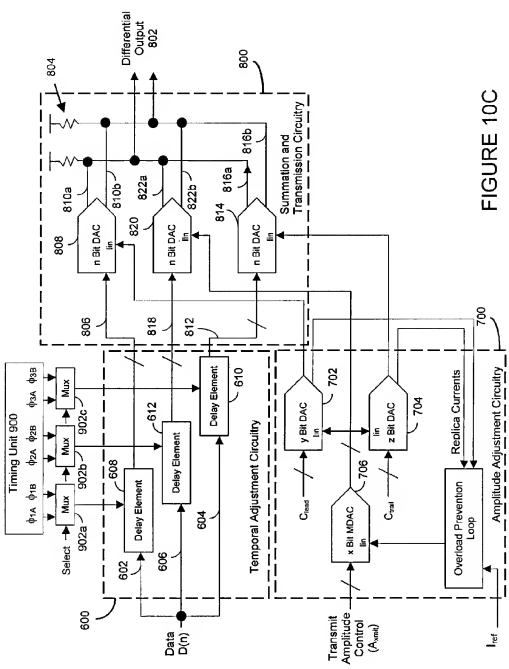

Differential Output 802
804
800
Summation and Transmission Circuitry
n Bit DAC
Timing Unit 900
Mux
Select
Delay Element
Temporal Adjustment Circuitry
y Bit DAC
z Bit DAC
x Bit MDAC
Overload Prevention Loop
Replica Currents
Amplitude Adjustment Circuitry
700
600
Data D(n)
Transmit Amplitude Control (A_cntrl)
I_ref

FIGURE 10C

FIGURE 11A

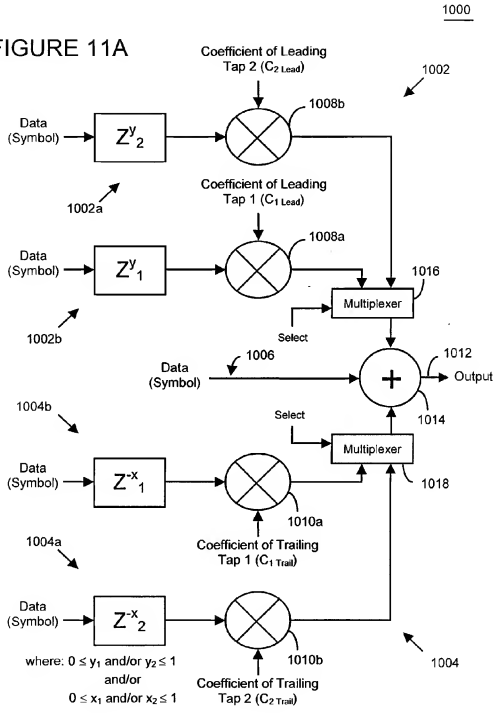
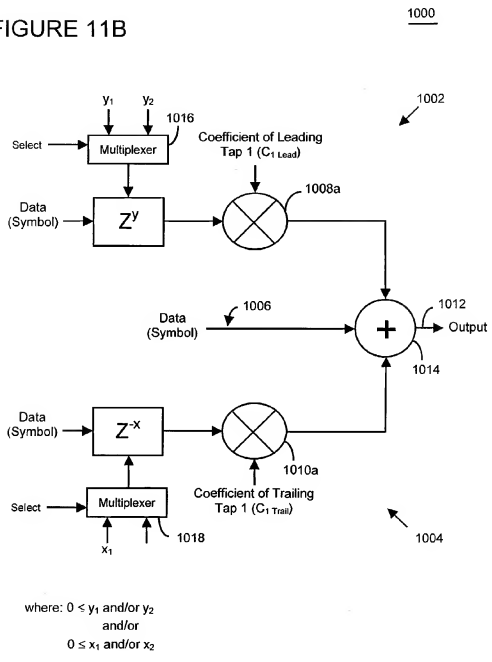

1000
Coefficient of Leading Tap 2 ($C_{2\ Lead}$)
1002
1008b
Data (Symbol)
Z^{y}_{2}
Coefficient of Leading Tap 1 ($C_{1\ Lead}$)
1002a
1008a
Data (Symbol)
Z^{y}_{1}
1016
Multiplexer
1002b
Select
1006
Data (Symbol)
1012
Output
1014
1004b
Select
Multiplexer
Data (Symbol)
Z^{-x}_{1}
1018
1010a
1004a
Coefficient of Trailing Tap 1 ($C_{1\ Trail}$)
Data (Symbol)
Z^{-x}_{2}
1010b
1004
where: $0 \le y_1$ and/or $y_2 \le 1$ and/or $0 \le x_1$ and/or $x_2 \le 1$
Coefficient of Trailing Tap 2 ($C_{2\ Trail}$)

FIGURE 11B



where: $0 \leq y_1$ and/or y_2
and/or
$0 \leq x_1$ and/or x_2

FIGURE 12A

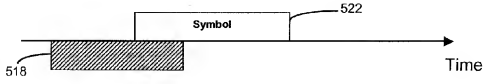


FIGURE 12B

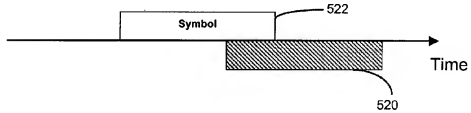


FIGURE 12C

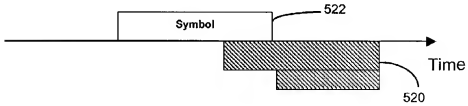


FIGURE 12D

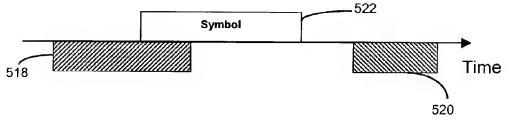


FIGURE 12E

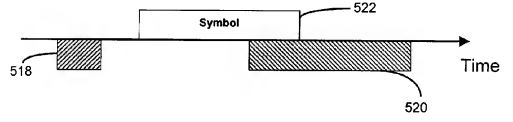


FIGURE 12F

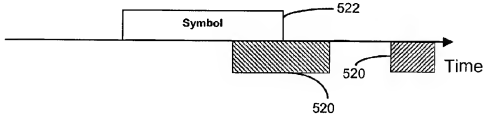


FIGURE 12G

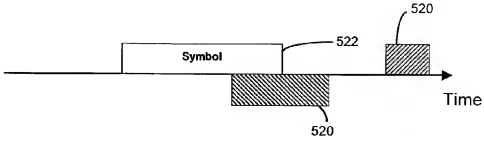
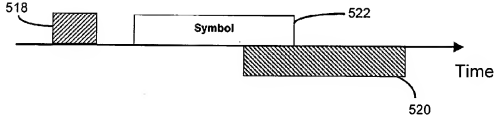


FIGURE 12H



SYSTEM AND METHOD OF EQUALIZATION OF HIGH SPEED SIGNALS

BACKGROUND OF THE INVENTION

[0001] This invention relates to a system and method for providing high-speed digital communications through a communications channel, and more particularly to an equalizer for communications systems implemented in wired type environments, for example microstrip, stripline, printed circuit board (e.g., a backplane) and cable.

[0002] Communications systems are continuing to increase the rate at which data is transmitted between devices. The increase in data rate presents a challenge to maintain, enhance or optimize the ability to recover the transmitted signal and thereby the information contained therein. Thus, in general, increasing the rate of transfer of data tends to adversely impact the fidelity of the received signal.

[0003] For example, high-speed digital baseband communications systems often encounter debilitating signal reflections and signal dispersion as the rate of data transfer increases. Signal reflections are often due to mismatches of impedances between the impedances of the devices (whether the devices are connected in a point-to-point or bus configuration), and/or mismatches of the impedance of the communications channel and termination resistors.

[0004] Signal dispersion, also known as intersymbol interference, may be caused by bandwidth limitations of the communications channel. Dispersion in many situations is due to two primary effects, namely, dielectric loss and skin effect. The effects of dielectric loss are often limited or minimized in communications systems, for example, wired systems, through careful design of the conductor insulator.

[0005] In backplane environments, however, intersymbol interference caused by dielectric loss may be difficult to eliminate, or sufficiently or adequately limit or minimize. In this regard, dispersion of the transmitted signal may be observed in the time domain as a symmetric broadening of that signal. This broadening of the transmitted signal produces both pre-cursor (before the pulse peak of the signal) and post-cursor (after the pulse peak of the signal) intersymbol interference.

[0006] Dispersion effects attributable to skin-effect tend to be observed predominately as post-cursor intersymbol interference. Although the debilitating effects caused by skin-effect may not be dominant when compared to effects of dielectric loss, skin-effect may be a source of dispersion to be addressed at higher transmission rates.

[0007] In short, pre-cursor and/or post-cursor impairment, whether due to skin effect or dielectric loss, must be addressed as transmission rates increase.

[0008] With reference to **FIGS. 1 and 2**, conventional high speed digital baseband communications systems often employ circuitry, for example, a finite impulse response filter ("FIR filter"), in the receiver to equalize the transmitted signals in an effort to address, or compensate for the effects of dispersion and reflection of the transmitted signal and/or the sensitivity of the system to that distortion or reflection. Such circuitry typically includes a one or more taps having fixed or pre-programmed "positions" and coefficients. These taps are typically "trailer" or "trailing" taps and, as such, the equalization circuitry address only post-cursor signal distortion.

[0009] In pre-emphasis equalization implementations (i.e., equalization circuitry and techniques implemented in the transmitter), the equalization circuitry also includes "trailer" or "trailing" taps to provide an equalization signal that is produced after transmission of the information signal. The pre-emphasis equalization circuitry, like the equalization circuitry implemented in the receiver, is designed to address only post-cursor signal distortion.

[0010] For example, in the backplane environment, conventional communications systems employ an FIR filter having fixed or pre-programmed tap positions and coefficients in the transmitter. The taps are positioned to compensate for post cursor intersymbol interference.

[0011] Regardless of where the equalization circuitry is implemented, the duration of the equalization signal and the relative position or placement of the tap(s) of the equalization circuitry are selected or designed to avoid interference with the signal representative of the transmitted information. As such, at the transmitter, there is no temporal overlap between the equalization signal and the information signal. In this way, the equalization signal is less likely to interfere with the pulse peak of the transmitted signal (i.e., the symbol or data signal).

[0012] Thus, while conventional equalization techniques may address, or compensate for the effects of some of the bandwidth limitations and reflections in the system, the ability of such techniques to provide sufficient compensation for high-speed communications may be limited. Moreover, not only may the conventional techniques be unsuitable to provide adequate compensation for the debilitating affects on the integrity of the transmitted signal in high speed communication systems, but the operation and corresponding impact of conventional equalization circuitry may not be adjusted as the environment of those systems change (for example, due to changes in temperature, operating conditions, data rate, and device parameters due to, for example, aging). That is, after design and manufacture, conventional equalization circuitry and techniques have limited flexibility when implemented within a particular environment or an environment that varies over time. This may severely limit the usefulness of such equalization circuitry and techniques when implemented in environments that change dramatically over time.

[0013] Notably, incorporating more complex equalization circuitry, for example, an FIR filter having many taps, tends to add cost, complexity and power consumption to a transmitter, receiver and/or transceiver. In addition, conventional pre-emphasis tends to suffer from over-equalization at the boundaries of the symbol (data signal) and may exhibit large parasitic capacitances thereby degrading the performance of the system. Such over-equalization may impact successive symbols or data signals, thereby contributing to the debilitating effects of signal reflections and dispersion. Accordingly, there is a tendency to implement equalization circuitry and techniques having a minimum of complexity and taps; however, such circuitry and techniques often are unable to provide sufficient compensation for high-speed communications systems.

[0014] Thus, there is a need for improved digital communications systems and techniques in order to enhance the

performance of, for example, high-speed digital communication systems through a communications channel, for example a backplane. There is a need for improved equalization circuitry and techniques that are capable of compensating for bandwidth limitations and reflections, improving the signal integrity in high-speed communications, and overcoming many of the shortcomings of conventional circuitry and techniques.

SUMMARY OF THE INVENTION

[0015] In a first principal aspect, the present invention is directed to a technique of, and system for enhancing the performance of high-speed digital communications through a communications channel, for example a backplane. In one embodiment of this aspect of the present invention, a data communications system includes a transmitter having equalization circuitry to compensate for bandwidth limitations and reflections in high-speed digital communication systems. The equalization circuitry is designed, programmed and/or configured to introduce compensation that is ideally the inverse of the intersymbol interference caused by the communications channel in order to improve the signal integrity in high-speed communications and enhance the operation and performance of such systems. In this regard, the equalization circuitry may include leading and/or trailing taps to reduce, minimize, mitigate or effectively eliminate pre-cursor and/or post-cursor intersymbol interference due to, for example, bandwidth limitations and reflections in high-speed digital communication systems. The amount of intersymbol interference may be programmed, adjusted or controlled by varying the positioning of the tap(s), varying the coefficients of the tap(s), and/or varying the pulse durations of the tap(s) (that is, the pulse duration of the equalization signal attributed to the tap).

[0016] In a second principal aspect, the present invention is a system for providing data communication over a communications channel, wherein the communications channel includes a backplane. The system comprises a first transmitter and a first receiver, both coupled to the communications channel. The first transmitter includes equalization circuitry having at least one leading tap and at least one trailing tap wherein the at least one leading tap and the at least one trailing tap provide an equalization signals that temporally overlap with a data signal and wherein the transmitter outputs an equalized data signal. The taps of the equalization circuitry may include programmable coefficients, programmable positioning, and/or programmable pulse durations.

[0017] In one embodiment, the system further includes a back channel for transmitting information from the first receiver back to the first transmitter wherein the information is representative of the coefficients of the taps of the equalization circuitry in the first transmitter. The system may also include a second transmitter coupled, via a communications channel, to a second receiver. In this embodiment, the first receiver may calculate information which is representative of the equalization circuitry's coefficients (for example, intermittently or periodically) and the second transmitter may transmit that information to the second receiver. Thereafter, the information may be provided to the first transmitter for implementation by the equalization circuitry. The first transmitter may use the information to adjust, program, alter or vary the operation or response of the equalization circuitry.

[0018] In another embodiment, the taps of the equalization circuitry include programmable coefficients and wherein information which is representative of the programmable coefficients is provided to the system using an external interface.

[0019] It should be noted that the information provided via the back channel or external interface, for example, may be the actual positions of the tap(s), coefficients of the tap(s), and/or pulse durations of the tap(s). Alternatively, the information may be adjustments, modifications and/or changes to be made to the positions of the tap(s), the coefficients of the tap(s), and/or the pulse durations of the tap(s). Thus, information which is representative of the positions of the tap(s), the coefficients of the tap(s), and/or the pulse durations of the tap(s) is intended to define the actual positions (i.e., absolute values) of the tap(s), coefficients of the tap(s), and/or pulse durations of the tap(s), as well as information pertaining to adjustments, modifications and/or changes to be made to the positions (i.e., relative values) of the tap(s), the coefficients of the tap(s), and/or the pulse durations of the tap(s).

[0020] In another principal aspect, the present invention is a system for providing data communication over a communications channel, wherein the communications channel includes a backplane. The system of this aspect includes a first transmitter coupled to a first receiver via the communications channel. The first transmitter includes equalization circuitry having a first trailing tap that provides an equalization signal that temporally overlaps with a data signal, wherein the transmitter generates an equalized data signal using the equalization signal and the data signal. The first trailing tap may include a programmable coefficient, a programmable position, and/or a programmable pulse

[0021] In one embodiment of this aspect of the present invention, the system further includes a back channel for transmitting information which is representative of the coefficient of the tap of the equalization circuitry to the transmitter. In this regard, the system may include a second transmitter coupled to a second receiver via the communications channel. The first receiver may calculate information which is representative of the coefficient (for example, periodically) and the second transmitter may transmit the information to the second receiver. The first transmitter may use the information to adjust, program, alter or vary the operation or response of the equalization circuitry

[0022] In another embodiment, the equalization circuitry may also include a second trailing tap wherein the second trailing tap provides an equalization signal that temporally overlaps with a data signal. The transmitter may generate the equalized data signal using the equalization signals provided by the first and second trailing taps and the data signal.

[0023] In another embodiment, the equalization circuitry further includes a leading tap wherein the leading tap provides an equalization signal that temporally overlaps with a data signal. The transmitter may generate the equalized data signal using the equalization signal provided by the leading tap, the equalization signals provided by the first and second trailing taps, and the data signal.

[0024] In yet another principal aspect, the present invention is a method for equalization of data signals that are transmitted over a communications channel (for example, a backplane). The method may include generating a first

equalization signal using a leading tap wherein the first equalization signal temporally overlaps with a data signal, generating a second equalization signal using a first trailing tap wherein the second equalization signal temporally overlaps with the data signal, and generating an equalized data signal using the first and second equalization signals with the data signal. The method may further include transmitting the equalized data signal.

[0025] The method of this aspect of the invention may further include generating a third equalization signal using a third trailing tap wherein the third equalization signal temporally overlaps with the data signal. In contrast, the method may include generating a third equalization signal using a third trailing tap wherein the third equalization signal does not temporally overlap with the data signal.

[0026] In one embodiment, the method may include controlling the first and/or second equalization signals by controlling the position of the trailing tap, the coefficient of the trailing tap and/or the duration of the tap.

[0027] In another embodiment, the method includes receiving the equalized data signal using a receiver, generating information which is representative of the coefficient of the trailing tap(s) (either absolute or relative information), and providing the information which is representative of the coefficient of the trailing tap to the transmitter to control, adjust or modify the coefficient of the trailing tap.

BRIEF DESCRIPTION OF THE DRAWINGS

[0028] In the course of the detailed description to follow, reference will be made to the attached drawings. These drawings show different aspects of the present invention and, where appropriate, reference numerals illustrating like structures, components, circuitry and/or elements in different figures are labeled similarly. It is understood that various combinations of the structures, components, circuitry and/or elements other than those specifically shown are contemplated and within the scope of the present invention.

[0029] **FIG. 1** is a block diagram representation of a pre-emphasis two-tap FIR filter topology;

[0030] **FIG. 2** is an illustration of the filter tap timing, in relation to the data signal ("Symbol"), according to the conventional two tap filter topology of **FIG. 1**;

[0031] **FIG. 3** is a block diagram representation of an exemplary communications system including a transmitter and a receiver;

[0032] **FIG. 4** is a block diagram representation of transmitter/receiver pairs of an exemplary communications system;

[0033] **FIGS. 5A and 5B** are block diagram representations of topologies of equalization circuitry in accordance with certain aspects of the present invention;

[0034] **FIG. 6** is an illustration of the tap timing relationship, in relation to the data signal ("Symbol"), according to one aspect of the present invention;

[0035] **FIG. 7** is an illustration of the tap timing relationship, in relation to the data signal ("Symbol"), according to another embodiment of the present invention;

[0036] **FIG. 8** is an illustration of an exemplary tap timing and amplitude relationship, in relation to the data signal and the resulting composite signal to be transmitted, according to one embodiment of the present invention;

[0037] **FIGS. 9A, 9B, 9C, 9D, 9E and 9F** are illustrations of exemplary tap timing, amplitude (coefficient), and pulse width relationships, in relation to the data signal, according to certain aspects of the present invention;

[0038] **FIGS. 10A, 10B and 10C** are detailed block diagram representations of a three tap transmit equalizer, in conjunction with additional transmitter circuitry, according to certain embodiments of the present invention;

[0039] **FIGS. 11A and 11B** are block diagram representations of another embodiment of the present invention including a plurality of selectable tap topologies/responses of equalization circuitry in accordance with certain aspects of the present invention; and

[0040] **FIGS. 12A, 12B, 12C, 12D, 12E and 12F** are illustrations of exemplary tap timing and pulse width relationships, in relation to the data signal (i.e., symbol), according to certain aspects of the present invention.

DETAILED DESCRIPTION

[0041] The present invention is directed to a technique of, and system for enhancing the performance of high-speed digital communications through a communications channel, for example a backplane. In one embodiment of the present invention, a transmitter includes equalization circuitry and techniques to compensate for bandwidth limitations and reflections in high-speed digital communication systems. The circuitry and techniques of this embodiment is designed, programmed and/or configured to introduce intersymbol interference in order to improve the signal integrity in high-speed communications and enhance the operation and performance of such systems. The circuitry and techniques include leading and/or trailing taps to reduce, minimize, mitigate or effectively eliminate precursor and/or post-cursor intersymbol interference due to, for example, bandwidth limitations and reflections in high-speed digital communication systems. In this way, the equalization circuitry and techniques may reduce, minimize or eliminate non optimum (e.g., over-equalization) at the boundaries of the data signal (i.e., the symbol).

[0042] Implementing the circuitry and techniques of the present invention in the transmitter may have an advantage in that the equalization effect may include an infinite impulse response (IIR) equalization component that is created by the impulse response of the communications channel. The leader and/or trailer taps of the equalization circuitry are adjustable "corrections" applied to the transmitted symbol. These "correction" pulses are dispersed along with the transmitted symbol by the impulse response of the communications channel. As such, the longer the channel's impulse response, the longer the correction pulses. This may reduce or minimize the number of taps required (over a time-sampled equalizer implemented at the receiver) to achieve a desired response and thereby improve or enhance the bandwidth of the transmitter output.

[0043] The circuitry and technique according to one embodiment of the present invention may include pre-emphasis implemented as an analog FIR filter having one or

more taps positioned before and/or after the transmitted symbol. The taps are positioned to temporally overlap with the data signal to thereby introduced intersymbol interference. The amount of intersymbol interference may be programmed, adjusted or controlled by varying the positioning of the tap(s), varying the coefficients of the tap(s), and/or varying the pulse durations of the tap(s) (that is, the pulse duration of the equalization signal attributed to the tap).

[0044] In one aspect of the present invention, the positions of the tap(s), the coefficients of the tap(s), and/or the pulse durations of the tap(s) may be adjusted or controlled during operation of the transmitter. In this regard, the adjustment or control of the taps of the circuitry for equalization may be accomplished using an adaptive algorithm.

[0045] For example, tap weights or coefficients (values) may be determined or controlled in response to a conventional linear adaptive algorithm (for example, Least Mean Square, Recursive Least Square, and stochastic versions thereof) to provide enhanced or optimal reception (maximum eye-opening) at a receiver. In a preferred embodiment, a stochastic zero forcing algorithm may be employed to provide convergence (stochastic Least Mean Square). In this regard, the adaptive algorithm uses samples of the received signal provided by the receiver to force the edges of the symbol pulse or data signal towards zero. Such an algorithm may have a robust convergence behavior.

[0046] In another embodiment, the adjustment or control of the taps of the equalization circuitry may be by the user via programming either before or during operation of the transmitter. In this regard, the user may provide information representative of the positions of the tap(s), the coefficients of the tap(s), and/or the pulse durations of the tap(s) in order to change, enhance or optimize the performance of the transmitter, receiver and/or the communications system. This information may be the actual positions of the tap(s), the coefficients of the tap(s) and/or the pulse durations of the tap(s). This information may also be adjustments, modifications and/or changes to be made to the positions of the tap(s), the coefficients of the tap(s), and/or the pulse durations of the tap(s). Such information may be provided directly to the transmitter (to be implemented within the equalization circuitry) or to a controller to distribute to the transmitter or set of transmitters.

[0047] In another embodiment, the positions of the tap(s), the coefficients of the tap(s), and/or the pulse durations of the tap(s) may be pre-programmed or pre-set, for example, by permanently, semi-permanently or temporarily (i.e., until re-programmed) storing information which is representative of the positions of the tap(s), the coefficients of the tap(s), and/or the pulse durations of the tap(s) in an SRAM, DRAM, ROM, PROM, EPROM, EEPROM or the like (e.g., configuring the state of a certain pin or pins on the package). In this embodiment of the present invention, the information representative of the positions of the tap(s), the coefficients of the tap(s) and/or the pulse durations of the tap(s) may be stored or in, for example an SRAM, DRAM, ROM or EEPROM. In this way, the transmitter may access the memory to retrieve the necessary information during start-up/power-up, initialization or re-initialization.

[0048] As mentioned above, the information which is representative of the positions of the tap(s), the coefficients of the tap(s), and/or the pulse durations of the tap(s) may be the actual positions of the tap(s), the coefficients of the tap(s) and/or the pulse durations of the tap(s). Alternatively, this information may also be adjustments, modifications and/or changes to be made to the positions of the tap(s), the coefficients of the tap(s), and/or the pulse durations of the tap(s).

[0049] It should be noted that the memory used to store the information representative of the positions of the tap(s), the coefficients of the tap(s) and/or the pulse durations of the tap(s) may be comprised of discrete component(s) or may reside on the integrated circuit containing the transmitter, receiver or transceiver.

[0050] In another embodiment, the pre-set or preprogrammed positions of the tap(s), the coefficients of the tap(s) and/or the pulse durations of the tap(s) may be fine-tuned to enhance the system performance. In this regard, after (or during) the performance of an initialization or re-initialization process, the system may implement fine adjustments to the predetermined positions of the tap(s), the coefficients of the tap(s) and/or the pulse durations of the tap(s). The fine adjustments may be accomplished or implemented by the user or by an adaptive algorithm as mentioned above. In this way, convergence may be obtained more rapidly and/or a less complex adaptive algorithm may be implemented.

[0051] It should be noted that all techniques for determining, calculating, setting, and/or resetting the positions of the tap(s), the coefficients of the tap(s) and/or the pulse durations of the tap(s) (i.e., the pulse duration of the equalization signal(s)), whether now known or later developed, are intended to be within the scope of the present invention.

[0052] With reference to **FIG. 3**, in one aspect, the present invention is a high-speed digital communication system **10** including transmitter **100** (having equalization circuitry) and receiver **200**. Briefly, transmitter **100** is connected to receiver **200** via communications channel **300**, for example, a backplane. In one embodiment, transmitter **100** encodes and transforms a digital representation of the data into electrical signals. The transmitter **100** also transmits the signals to receiver **200**. The received signals, which may be distorted with respect to the signals transmitted into or onto communications channel **300** by transmitter **100**, are processed and decoded by receiver **200** to reconstruct a digital representation of the transmitted information.

[0053] With reference to **FIG. 4**, the digital communication system **10** typically includes a plurality of transmitters and receivers. In this regard, communications system **10** includes a plurality of unidirectional transmitter and receiver pairs (transmitter **100***a* and receiver **200***b*; and transmitter **100***b* and receiver **200***a*). Transmitter **100***a* and receiver **200***a* may be incorporated into transceiver **400***a* (in the form of an integrated circuit). Similarly, transmitter **100***b* and receiver **200***b* are incorporated into transceiver **400***b*. From a system level perspective, there are a plurality of such transmitter/receiver pairs in simultaneous operation, for example, four, five, eight or ten transmitter/receiver pairs, communicating across communications channel **300**. Thus, in operation, the transmitter and receiver pairs simultaneously transmit data across channel **300**.

[0054] In one embodiment, transmitters **100** and receivers **200** employ a multilevel pulse amplitude modulated (PAM-n) communications technique. For example, transmitters

100 and receivers **200** may employ a PAM-4 signaling technique to send two bits of data through channel **300**. That is, each transmitter/receiver pair may operate in the same manner to send two bits of data for each symbol transmitted through the channel **300**.

[0055] In one embodiment, five successive symbols are associated with each eight-bit data byte. The additional overhead associated with this form of encoding may be used to ensure adequate symbol crossings, necessary for timing recovery and/or to provide DC balance. That overhead may also be used to transmit control information for controlling or modifying the operation of the adaptive or adjustable equalization circuitry in a transmitter.

[0056] It should be noted that although certain aspects of the present invention are described in the context of PAM-4 signaling techniques (for example, the multiplying digital to analog converters (MDACs) in **FIG. 10A**), the present invention may utilize other modulation formats that encode fewer or more bits per symbol codes based on other than byte wide user may be readily adapted or employed. Moreover, other communications mechanisms that use different encoding tables, other than four levels, or use other modulation mechanisms may also be used, for example, PAM-5, PAM-8, PAM-16, CAP, and wavelet modulation. In this regard, the techniques described herein are in fact applicable to any and all modulation schemes, including but not limited to, PAM-4 encoding described herein.

[0057] As mentioned above, in one embodiment, equalization circuitry is incorporated into each transmitter of a high-speed digital communication system. In this embodiment, the equalization circuitry incorporates leading and/or trailing taps to reduce, minimize, mitigate or effectively eliminate pre-cursor and/or post-cursor intersymbol interference due to, for example, bandwidth limitations and reflections in high-speed digital communication systems. The extent of equalization introduced by the circuitry, however, may vary between equalization circuitry of each transmitter in the high-speed digital communication system.

[0058] With reference to **FIG. 5A**, in one embodiment, equalization circuitry **500** may be characterized as an analog FIR filter having a basic topology including three taps, namely leading tap **502**, trailing tap **504** and information, symbol or data tap **506**. The leading and trailing taps **502** and **504**, respectively, are designed and configured to introduce compensation that is ideally the inverse of the effects caused by the communications channel. The positioning of the taps, relative to the data or symbol tap **506**, may be controlled, varied or adjusted to provide a desired or predetermined output. The positions of taps **502** and **504** are determined and/or adjusted by changing the values of y_1 and x_1, respectively.

[0059] The amplitude of the pre-cursor and post-cursor intersymbol interference is determined or controlled by the respective coefficients (C_{lead} and C_{trail}). In particular, C_{lead} determines the amplitude of the signal introduced in the leading tap and C_{trail} determines the amplitude of the signal introduced in the trailing tap. A scaling operation is performed whereby the leading or trailing signals are "multiplied" by C_{lead} and C_{trail}. This operation is represented by the multiplier symbol **508** and **510**, respectively. Both C_{lead} and C_{trail} may be controlled, set and/or changed as described above.

[0060] A scaling operation may also be performed with respect to the symbol or data signal. In one embodiment, data or symbol is "multiplied" by C_{symbol}. It should be noted, however, that the coefficient C_{symbol} is typically 1.

[0061] With continued reference to **FIG. 5A**, the output of the leading tap **502** and trailing tap **504** are combined, along with the output of data or symbol tap, via a summation operation (i.e., summation symbol **516**) to produce an output signal **514**. The output signal **514** is representative of the signal transmitted by the transmitter to the receiver.

[0062] As mentioned above, the amount of equalization may be adjusted or controlled by altering the positioning of the tap(s), changing the coefficients of the taps, and/or varying the pulse durations of the tap(s). With reference to **FIGS. 6 and 7** and continued reference to **FIG. 5A**, a change in the positioning of the taps will produce a change in the timing of signals **518** and **520** generated by the leading and trailing taps **502** and **504**, respectively. The ability to change, vary or program the positions of the taps is illustrated by lines **524***a* and **524***b* (signal **518**), and lines **524***c* and **524***d* (signal **520**).

[0063] With continued reference to **FIGS. 5A, 6** and **7**, a change in the coefficients of the taps will produce signals **518** and **520** having larger or smaller amplitudes. The amplitude of equalization signal **518** (of leading tap **502**) may be characterized as $D(n)\times C_{lead}$. The amplitude of equalization signal **520** (of trailing tap **504**) may be characterized as $D(n)\times C_{trail}$. Lines **526***a* and **526***b* indicate the capability of changing, varying and/or programming the coefficients of the taps.

[0064] Changing the pulse duration of signals **518** and **520** will, among other things, increase or decrease the high frequency equalization. In this regard, the extent of high frequency equalization increases as the duration of the equalization signals **518** and **520** decrease. Under this circumstance, higher speed analog and digital circuitry may be necessary to implement the transmitter. It should be noted that the ability to change the pulse duration of the taps is also illustrated by lines **524***a* and **524***b* (leader equalization signal **518**), and lines **524***c* and **524***d* (trailer equalization signal **520**).

[0065] With reference to **FIG. 5B**, in another embodiment, the equalization circuitry **500** may include a plurality of leading taps **502** and/or a plurality of trailing taps **504**. As described above, the extent of pre-cursor and post-cursor correction (i.e., intersymbol interference) may be adjusted or controlled by altering the positioning of the tap(s) of leading taps **502** and trailing taps **504**, changing the coefficients of the taps of leading taps **502** and trailing taps **504**, and/or varying the pulse durations of the tap(s) (that is, the pulse duration of the equalization signal attributed to the tap(s)) of the leading taps **502** and trailing taps **504**.

[0066] It should be noted that a digital signal generator ((for example, a signal generator having a variable, controllable or programmable period)) may be employed to provide equalization signals having pulse durations and/or shapes that are different than the period and/or shape of the symbol. The digital signal generator may generate signals having different pulse widths and as such, equalization signals may have different widths and each may be different than the period of the symbol. The signal generator may supply the input(s) to the tap(s) of the leading tap(s) and/or the trailing taps.

[0067] It should be noted that there are other techniques for controlling, programming or adjusting the duration of equalization signals **518** and **520**, as described below.

[0068] In one embodiment of the present invention, the positions of the tap(s), the coefficients of the tap(s), and/or the pulse durations of the tap(s) are adjusted or controlled during operation of the transmitter, for example using an adaptive algorithm, as described above. Moreover, the enhanced, suitable or optimum values of the positions of the tap(s), the coefficients of the tap(s), and/or the pulse durations of the equalization signal(s) may be determined by heuristic means to provide the range of equalization (after adaptation) for a given class of communication channels.

[0069] In another embodiment, the adjustment or control of the taps of the equalization circuitry may be user programmable (either before or during operation of the transmitter). The positions of the tap(s), the coefficients of the tap(s), and/or the pulse durations of the equalization signal(s) may, in another embodiment, be pre-programmed or pre-set, for example, by permanently, semi-permanently or temporarily (i.e., until re-programmed) storing information representative of the positions of the tap(s), the coefficients of the tap(s), and/or the pulse durations of the tap(s) in an SRAM, DRAM, ROM, PROM, EPROM, EEPROM or the like. The pre-programmed or pre-set positions of the tap(s), the coefficients of the tap(s) and/or the pulse durations of the tap(s) may thereafter be finetuned to enhance the system performance.

[0070] In those instances where the positions of the tap(s), the coefficients of the tap(s), and/or the pulse durations of the tap(s) are adjusted or controlled using an adaptive algorithm, a back channel may be advantageous to provide information to the equalization circuitry that resides in the transmitter. A back channel is a communication path and technique that facilitates providing information that is determined and/or generated by a receiver and used by the corresponding transmitter to adjust or modify output signal characteristics of that transmitter. Thus, with reference to **FIG. 4**, equalization circuitry that resides in transmitter **100***a* receives control or adaptation information that is determined and/or generated by receiver **200***b* via transmitter **100***b* and receiver **200***a*. Here, back channel is comprised of transmitter **100***b*, receiver **200***a*, and communications channel **300** that connects transmitter **100***b* to receiver **200***a*.

[0071] In one embodiment, the back channel forms a part of the user data channel. In this way, back channel data may be transmitted asynchronously at the same time user data is transmitted without reducing or significantly impacting the amount of channel communications capacity dedicated to user data. Thus, in one embodiment, back channel data is sent in a back channel data frame or data packet. The data frame may include a frame header, a set of data bits, a set of control bits, and a data frame trailer. The frame header is used to mark the beginning of a data frame to allow the transmitter and receiver to remain synchronized as to the proper beginning of the data frame. The set of data bits contains the data to be transmitted across the back channel and is distinguished from the set of control bits used to control the operation of the back channel as necessary. Finally, the data frame trailer is used to mark the end of a data frame to further allow the transmitter and receiver to remain synchronized as to the proper end of the data frame.

[0072] It should be noted that other message formats and features, such as error correction or detection, may be implemented in the back channel frame. Indeed, any and all formats, whether now known or later developed, are intended to be within the scope of the present invention. Moreover, it is possible to construct an arbitrarily complex frame for the back channel information and have the frame carried by the sub channel described herein.

[0073] It should be further noted that the information provided via the back channel may be the actual positions (absolute values) of the tap(s), coefficients of the tap(s), and/or pulse durations of the tap(s). Alternatively, the information may be adjustments, modifications and/or changes to be made to the positions (relative values) of the tap(s), the coefficients of the tap(s), and/or the pulse durations of the tap(s). Thus, information which is representative of the positions of the tap(s), the coefficients of the tap(s), and/or the pulse durations of the tap(s) is intended to define the actual positions of the tap(s), coefficients of the tap(s), and/or pulse durations of the tap(s), as well as information pertaining to adjustments, modifications and/or changes to be made to the positions of the tap(s), the coefficients of the tap(s), and/or the pulse durations of the tap(s).

[0074] With reference to **FIGS. 5A and 7**, the equalizer tap timing relationship where Y_1 and x_1 are both equal to about ½ produces leading and trailing equalization signals **518** and **520** that encompass about the entire period of the symbol of information signal. The leading and trailing equalization signals **518** and **520**, respectively, each temporally overlap symbol signal **522** by appropriately 50% of the period of symbol signal **522**. Unlike a conventional equalization circuitry and techniques where the temporal overlap (at the transmitter) is 0%, the intersymbol interference may reduce or eliminate over-equalization at the symbol boundaries.

[0075] As mentioned above, the output of equalization circuitry **500** may, as a practical matter, be determined adding leading and trailing equalization signals **518** and **520**, respectively, and symbol signal **522**. For example, where y_2 and x_2 are both equal to about ½, and the coefficients of the leading and trailing taps **502** and **504** produce equalization signals **518** and **520** as illustrated in **FIG. 8**, output signal **528** is produced. As the positions of the tap(s), the coefficients of the tap(s), and/or the pulse durations of the tap(s) vary, the shape and energy content of output signal **528** will change accordingly.

[0076] For given environments, it may be advantageous to implement equalization circuitry according to the present invention that includes taps having certain or predetermined positions, coefficients, and pulse durations. The specific delays may be determined by heuristic means to provide the "best" range of equalization (after adaptation) for a class of communication channels. With reference to **FIG. 9A**, in one embodiment, the leading and trailing taps are arranged to provide an equalization (pulse) signal overlap of about 25% of the data signal (i.e., symbol).

[0077] In another embodiment, the duration(s) or period(s) of the "taps" may be selected to be other than a full period of the symbol or data signal. With reference to **FIG. 9B**, the period or duration of the equalization signals is about ½ of the period of the symbol or data signal. This embodiment may enhance high frequency equalization relative to the

equalization provided by **FIGS. 7 and 9A**. As such, a system implementing an equalization timing relationship are illustrated in **FIG. 9B**, may require higher speed analog and digital circuitry relative to the systems implementing **FIGS. 7 and 9A**.

[0078] In addition, as mentioned above, the positions of the tap(s), the coefficients of the tap(s), and/or the pulse durations of the tap(s) determine the shape and energy content of output signal **528**. Certain communications environments may require differing amounts of post-cursor and/or pre-cursor correction in order to provide a desired response, for example, as measured by the receiver. In one embodiment of the present invention, the trailing taps may include more than one tap each having a position, coefficient and/or pulse duration that produces a relative tap timing and amplitude to "correct" for a post-cursor distortion that is larger than pre-cursor distortion. With reference to **FIGS. 9C, 9D** and **9E**, a second tap of trailing taps **504** may provide additional equalization signal to address this disproportionate distortion. The timing responses of **FIGS. 9C, 9D** and **9E** may be generated, for example, using the topology of **FIG. 5B**.

[0079] Moreover, the taps of trailing taps **504** may have different positions, coefficients and/or pulse durations. In this way, the tap of trailing taps **504** may provide additional equalization signal to address distortions that vary greatly over time.

[0080] Similarly, in certain circumstances, it may be advantageous to implement equalization circuitry that produces a relative tap timing and amplitude relationship as illustrated in **FIG. 9F**. In this regard, a second tap of leading taps **502** and a second tap of trailing taps **504** may provide additional equalization signal to address pre-cursor and post-cursor distortions that exhibit consideration variation. The pulse duration of equalization signal generated by the second taps of the leading and trailing taps are shorter than that of the first taps. As such, the correction signals produced by the leading and trailing taps **502** and **504** of the equalization circuitry may address large variations of precursor and post-cursor distortions. These variations may enhance the ability of the receiver to repeatedly recover the transmitted signal.

[0081] It should be noted that the timing response of **FIG. 9F** may be generated using the topology of **FIG. 5B**.

[0082] With reference to **FIG. 10A**, in one embodiment of the present invention, the equalization circuitry **500** may include temporal adjustment circuitry **600**, amplitude adjustment circuitry **700**, and summation and transmission circuitry **800**. The temporal adjustment circuitry **600** is designed to provide or create, among other things, the selected timing relationship between the pre-equalized data signal, leading tap signal and the trailing tap signal. In this regard, the temporal adjustment circuitry receives leading tap input **602**, trailing tap input **604** and data (i.e., symbol) input **606**. Each input is applied to a corresponding shift register, which is selectively clocked to generate the selected or desired timing relationships between the equalization signals and the symbol (i.e., the pre-equalized data signal). It should be noted that, in this embodiment, leading tap input **602**, trailing tap input **604** and data input **606** are the same signal.

[0083] With continued reference to **FIG. 10A**, in particular, shift register **608** receives leading tap input **602** and, in response to the timing signal ϕ_1, outputs a delayed and temporally aligned version of input **602**. Similarly, shift register **610** receives trailing tap input **604** and, in response to the timing signal ϕ_3, outputs a delayed temporally aligned version of input **604**. Shift register **612** receives the symbol or pre-equalized signal and, in response to the timing signal ϕ_2, outputs a delayed and temporally aligned version of input **606**. The signals output by the shift registers are precisely aligned relative to each other.

[0084] It should be noted that one, some or all of shift register **608, 610**, and **612** may be comprised of a set or group of parallel latches (the number in the set or group being equal to the number of digital inputs **602, 604**, and **606**). In addition, one, some or all of shift register **608, 610**, and **612** may be comprised a plurality of cascaded sets or groups of parallel latches (the number of latches in each set or group being equal to the number of digital inputs **602, 604**, and **606**) in order to introduce a "fixed" delay (i.e., the delay through the set(s) or group(s) of parallel latches) of the signals relative to the other shift registers. Thus, in this embodiment, the relative delay of input signals **602, 604** and **606** introduced by the shift registers and timing unit is equal to the delay introduced by the design of the respective shift registers and the differences in phases/delays between timing signals ϕ_1, ϕ_2, and ϕ_3.

[0085] Moreover, other circuitry, for example, flip/flops and/or pass gates (whose gate electrodes are coupled to the timing signals from timing unit **900**), may be implemented to provide a delayed and temporally aligned version of an input. Indeed, any digital delay element having a controllable delay, whether now know or later developed, may be implemented in the present invention to delay and temporally align the symbol and equalization signals.

[0086] With continued reference to **FIG. 10A**, the timing unit **900** provides a plurality of timing or clock signals having highly precise phase (and frequency) relationships. The timing unit **900** may be well known clock alignment circuitry, for example, delay locked loop circuitry or phase locked loop circuitry. As such, a highly precise phase (and frequency) relationship between data input **606** and leading and trailing tap inputs **602** and **604**, respectively, may be sustained during operation of the system and over a wide operating environment.

[0087] It should be noted that other clock alignment circuitry may also be employed. Indeed, any clock alignment circuitry that outputs a plurality of clock signals having highly precise and constant phase (and frecuency) relationships, whether now know or later developed, may be implemented in the present invention.

[0088] The temporal adjustment circuitry **600** and timing unit **900** may also provide a predetermined and/or programmable relationship between the duration of the leading tap signal, the trailing tap signal, and/or the pre-equalized data signal (i.e., symbol). In this regard, in one embodiment, the clock signals used to clock the shift registers may be selected to provide a predetermined and/or programmable relationship between the signals. For example, the period of timing signal ϕ_1 may control the period or duration of leading tap signal **602**. In this way, a precise period or duration of the leading tap signal (relative to the pre-equalized signal and/or trailing tap signal) may be selected to provide a desired amount of pre-cursor equalization.

[0089] Similarly, the period of timing signal ϕ_2 may control the period or duration of trailing tap signal **604**. Thus, a precise duration of the trailing tap signal (relative to the pre-equalized signal and/or leading tap signal) may be selected to provide a desired amount of post-cursor equalization.

[0090] Moreover, the period of the leading and trailing tap signals **602** and **604** may be programmed or controlled independently. That is, each tap signal may be programmed to a unique or different period or duration. In this way, the communication system has a considerable degree of flexibility in addressing pre-cursor and post-cursor distortions.

[0091] With continued reference to **FIG. 10A**, the amplitude adjustment circuitry **700** is designed to provide predetermined and/or programmable adjustment of the amplitude of the leading and trailing tap signals in relation to the transmitted signal (A_{xmit}). In this regard, multiplying digital to analog converters ("MDACs") **702** and **704** convert the tap coefficients C_{lead} and C_{trail} into analog representations thereof. The MDACs **702** and **704** use the analog representation of the transmit amplitude control (A_{xmit}) as a reference current to provide appropriate scaling.

[0092] The appropriately scaled analog representations of tap coefficients C_{lead} and C_{trail} are provided to the summation and transmission circuitry **800**. In particular, MDACs **702** and **704** convert the digital C_{lead} and C_{trail} tap coefficients into a current to drive for MDACs **808** and **814**, respectively, in summation and transmission circuitry **800**. The resolution of MDACs **702** and **704** may be selected according to the anticipated range of the leading and trailing coefficients. In this regard, in certain environments, the amplitude of the equalization signal of the trailing tap(s) may be larger than that of the leading tap(s) in order to address disproportionate post-cursor dispersion. Accordingly, it may be advantageous to employ a higher resolution DAC to convert the trailing coefficient.

[0093] As mentioned above, the leading and trailing taps provide intentional corrective intersymbol interference. This, as the coefficients of the leading and trailing taps are varied to provide enhanced or optimum system performance, the peak amplitude of the transmitted signal will also vary. Accordingly, it may be advantageous to incorporate an additional "control loop" to limit, minimize or prevent overdriving the transmitter output in response to variations of the leading and trailing tap coefficients and thereby prevent clipping of the signal at the transmitter output.

[0094] With continued reference to **FIG. 10A**, in a preferred embodiment, replica current(s) from coefficient MDACs **702** and **704** are compared to the reference current to generate the transmitter output current level. In one embodiment, this may also be the reference level for all symbol and coefficient DACs **702**, **704** and **706**. As such, for a given timing and range of coefficient taps, an algorithm may be implemented to prevent the clipping of the output signal.

[0095] It should be noted that there are many methods which may be utilized to provide an automatic gain control (AGC) technique, for example, the output of a peak detector may be compared to a reference signal (i.e., the maximum allowable transmit amplitude) and, using a standard control loop (e.g., a lossy-type integrator) a reference voltage may be determined for DAC **706**. Indeed, all techniques to prevent clipping of the output as a result of varying the tap coefficients and varying the pulse overlapping of the taps and the symbol (as a result of the timing relationship of the correction taps relative to the transmitted symbol or data and/or the pulse durations of the equalization signals), whether now know or later developed, are intended to be within the scope of the invention.

[0096] The summation and transmission circuitry **800** is designed to sum the equalization signals and the pre-equalized data signal into an equalized data signal and output the equalized data signal (differential output **802**) onto the communications channel, for example, a backplane. In this regard, summation and transmission circuitry **800** receives the temporally aligned leading tap, trailing tap, and data signals from temporal adjustment circuitry **600**. Using the amplitude information from the amplitude adjustment circuitry **700**, the summation and transmission circuitry **800** generates a differential output **802**. In one embodiment, the transmitted signal (i.e., differential output **802**) is generated via current summation and thereafter translated into a voltage using termination resistors **804**.

[0097] In particular, the temporally aligned leading tap signal **806** is provided to MDAC **808**. In response, MDAC **808** converts signal **806** to an analog representation thereof to produce leading tap signals **810***a* and **810***b*. The output of MDAC **702** provides the appropriate scaling of the leading tap correction (leading tap coefficient C_{lead}) by converting the leading tap coefficient C_{lead} to an analog signal and providing that signal to MDAC **808**. In one embodiment, MDAC **808** uses the analog representation of the leading tap coefficient C_{lead} as the reference current to provide appropriate scaling based on the magnitude of coefficient C_{lead}.

[0098] Similarly, temporally aligned trailing tap signal **812** is provided to MDAC **814** to generate the leading tap signals **816***a* and **816***b*. The MDAC **820** receives the analog representation of the trailing tap coefficient (from MDAC **704**) to introduce or implement the appropriate scaling of the trailing tap correction. In one embodiment, MDAC **814** uses the analog representation of the leading tap coefficient C_{trail} as a reference current to provide appropriate scaling for MDAC **814** based on coefficient C_{trail}.

[0099] Finally, the temporally aligned data signal **818** is provided to MDAC **820** which converts the digital information to an analog representation thereof. In one embodiment, MDAC **820** uses the analog representation of the transmit amplitude control A_{xmit} as a reference current to provide appropriate scaling. The analog representation is output on signal lines **822***a* and **822***b*.

[0100] As mentioned above, the outputs of the parallel combination of MDACs **808**, **814** and **820** are summed via current summation and translated into a voltage signal using termination resistors **804** (see, for example, signal **528** in **FIG. 8**). The termination resistors **804** are employed as a current summation node. In this way, the output exhibits a low impedance and the bandwidth of the output is correspondingly high. While more than three taps may be implemented, increasing the number of taps may impact the bandwidth of the transmitter or the system due to parasitic capacitances introduced by the transmitter.

[0101] It should be noted that the embodiment illustrated in **FIG. 10A** may be well suited for implementing PAM-4

baseband transmission. In this regard, parallel current MDACs are employed to convert the binary data D(n) to PAM-4. However, by changing the configuration of the MDACs other baseband modulation schemes may be implemented, for example, a one-bit MDAC may be used for NRZ transmission scheme. In this regard, with reference to **FIGS. 10B** and **10C**, under those circumstances where the transmission or encoding scheme is a multilevel pulse amplitude modulated (PAM-n) communications technique, MDACs **808**, **814**, and **820** may be n-level MDACs to accommodate that transmission or encoding scheme.

[0102] With continued reference to **FIG. 10A**, under those circumstances where the pulse duration of equalization signals changes (or is modified) to, for example, enhance or optimize the performance of the system, different pulse durations of leading tap input signals **602** and trailing tap input signal **604** may be provided by adjusting the period of timing signals ϕ_1 and ϕ_2. In this regard, a longer period (or larger pulse duration) may be obtained by applying timing signals to shift registers **608** and **610** that have a longer period. In this way, the value that resides in the shift registers is maintained in those registers for a longer period of time relative to symbol or data signal **606**. Notably, the timing signals ϕ_1 and ϕ_2 will maintain a precise phase relationship with respect to timing signal ϕ_2 regardless of changes or variations in the period(s) of ϕ_1 and ϕ_2.

[0103] Similarly, by selecting and applying timing signals ϕ_1 and ϕ_2 that have a shorter period (or shorter pulse duration), the equalization signals will have a shorter period relative to symbol or data signal **606**. In this way, the value that resides in the shift registers is clocked-out of the registers more quickly (relative to symbol or data signal **606**) and the digital information (i.e., **806** and **812**) provided to MDACs **808** and **814** changes accordingly so that the lengths of correction signals are shorter (compare, for example, **FIGS. 9A and 9B**).

[0104] The appropriate timing signals may be provided by timing unit **900** and may be selected or programmed based on the considerations described above. That is, in one embodiment, the pulse durations of the tap(s) are adjusted or controlled during operation of the transmitter, for example using an adaptive algorithm, as described above. In another embodiment, the adjustment or control may be by user programming (either before or during operation of the transmitter). The pulse durations of the tap(s) may, in another embodiment, be pre-programmed or pre-set, for example, by permanently, semi-permanently or temporarily (i.e., until re-programmed) storing information representative of the pulse durations of the tap(s) in an SRAM, DRAM, ROM, PROM, EPROM, EEPROM or the like. The pre-programmed or pre-set pulse durations of the tap(s) may thereafter be fine-tuned to enhance the system performance.

[0105] It should be noted that there are many techniques for generating equalized data signal **802**. For example, the summation of the equalization signals and the data signal (symbol) may be performed before converting the digital signals to analog representations. Under this circumstance, one DAC may be employed rather than three as illustrated in **FIGS. 10A, 10B** and **10C**.

[0106] With reference to **FIGS. 11A and 11B**, in another embodiment of the present invention, a plurality of topologies of the equalization circuitry may be predetermined, preset or pre-programmed and selection of the topology to be implement by the equalization circuitry may be made, for example, at start-up/power-up or may be made during installation of the system (via hardwiring a pin on the package to a given state). In this embodiment, the plurality of topologies may have: (1) predetermined, preset or pre-programmed timing relationships of the correction taps relative to the transmitted symbol (data signal), (2) predetermined, preset or pre-programmed tap coefficients, and/or (3) predetermined, preset or pre-programmed pulse durations of the equalization signals. A given topology of the plurality of topologies may be selected by the user according to, for example, an anticipated response of the system based on the communications environment. In this way, certain aspects of the complexity may be reduced or eliminated.

[0107] With reference to **FIG. 11A**, in one embodiment, leading taps **1002** may include two tap paths **1002***a* and **1002***b* (each path may have one or more taps). The selection of leading tap path **1002***a* or **1002***b*, and as such the leading tap path to be incorporated into the equalization process, is made by multiplexer **1016**. It should be noted that leading taps **1002** may include more than just two tap paths.

[0108] Similarly, trailing taps **1004** may include two tap paths **1004***a* and **1004***b* (each path may have one or more taps). The selection of leading tap path **1004***a* or **1004***b*, and as such the trailing tap path to be incorporated into the equalization process, is made by multiplexer **1018**. As with the leading taps, trailing taps **1004** may include more than just two tap paths.

[0109] In one embodiment, for example, first leading tap path **1002***a* and trailing tap path **1004***a* may provide a tap timing, amplitude (coefficient), and pulse width relationship according to **FIG. 7** or **FIG. 9A**. A second leading tap path **1002***b* and trailing tap path **1004***b* may provide a tap timing, amplitude (coefficient), and pulse width relationship according to **FIG. 9B** or **FIG. 9C**.

[0110] With reference to **FIG. 11B**, the selection of the topologies is implemented using multiplexers **1016** and **1018**. Here, however, the tap timing(s) of leading and trailing taps **1002** and **1004** are controlled by the multiplexers. This embodiment may be implemented in a manner according to **FIG. 10C**. In this regard, multiplexers **1016** and **1018** of **FIG. 11B** are represented by multiplexers **902***a* and **902***c* of **FIG. 10C**. Multiplexer **902***b* incorporates more flexibility into system **10** in that all three taps may be temporally adjusted.

[0111] With continued reference to **FIG. 11B**, the topologies of the leading and trailing taps may or may not include intersymbol interference (i.e., temporal overlap of the equalization signals with the data (symbol) signal). For example, a first topology may include y_1, and x_1, that are equal to about ½ or ¾. A second topology may include y_2 and/or x_2 that are greater than or equal to about 1. Thus, while the first topology includes temporal overlap between the leading/trailing taps and the data signal (symbol) (as illustrated in, for example, **FIGS. 7, 9A** and **9B**), the second topology will include no temporal overlap or only the leading or trailing tap will temporally overlap.

[0112] The user or system may select between the predetermined equalization effects (via multiplexers **1016** and **1018** and select signals) at installation (via configuring the

state of a certain pin or pins on the package), after start-up/power-up, during an initialization sequence or a re-initialization sequence, or during operation. The predetermined, preset or pre-programmed parameters of the taps may be "hardwired" permanently, semi-permanently or temporarily (i.e., until re-programmed) by way of a DRAM, SRAM, ROM, PROM, EPROM, EEPROM or the like (e.g., configuring the state of a certain pin or pins on the package).

[0113] In one embodiment, a register on the transmitter may be programmed to store information representative of the selected leading and/or trailing tap paths. The register may be programmed or accessed after start-up/power-up, during an initialization sequence or a re-initialization sequence, or during operation. In this embodiment, the selection of the tap paths to be implemented in the equalization process may be made by the user or the system, via programming of the register, according to, for example, a given or anticipated response of the communications channel. In this regard, a given tap path may be advantageous in systems that experience or observe considerable jitter of the clock and/or data signals or in systems that experience or observe considerable crosstalk between signal lines. As such, the selected response of the equalization circuitry may be implemented via programming of a discrete register or a register that may reside on the integrated circuit containing the transmitter (and the equalization circuitry).

[0114] In one embodiment, the parameters of the taps may also be fine tuned to enhance the system performance. In this regard, after (or during) the performance of an initialization or re-initialization process, the system may implement fine adjustments to the predetermined, preset or pre-programmed timing relationships of the correction taps, tap coefficients, and/or pulse durations of the equalization signals. The fine adjustments to these parameters may be accomplished using any of the techniques described above. Indeed, all techniques for determining these parameters, whether now known or later developed, are intended to be within the scope of the present invention.

[0115] Thus, in one embodiment of this aspect of the present invention, a first path of the leading and/or trailing taps may be "turned off" or disabled and another leading and/or trailing tap path may be "turned on" or enabled quite rapidly. In this embodiment, the system need not determine the initial parameters (or some of the initial parameters) of the tap paths because such parameters may be stored in, for example, a DRAM, SRAM, ROM or EPROM. As such, the time delay in implementing a new equalization response or effect may be reduced and the system may become operational more rapidly.

[0116] It should be noted that more than two leading tap paths and/or two trailing tap paths may be implemented. Indeed, the equalization circuitry may include as many leading and/or trailing tap paths as desired. In each of such tap paths, the taps may include a predetermined, preset or pre-programmed timing relationship, coefficients, and/or pulse durations. The user or system may then select from the plurality of tap paths which provides a given, desired or appropriate response.

[0117] The system according to one aspect of the present invention includes leading and/or trailing tap(s) to provide intentional corrective intersymbol interference to reduce, minimize, mitigate or effectively eliminate pre-cursor and/or post-cursor intersymbol interference due to, for example, bandwidth limitations and reflections in high-speed digital communication systems. The amount of intersymbol interference correction designed into the equalization topology is controlled or determined by the timing relationship of the correction taps relative to the transmitted symbol or data, the coefficients of the taps, and/or the pulse durations of the equalization signals.

[0118] The equalization structure and techniques of the present invention are incorporated into the transmitter. As mentioned above, implementing the equalization circuitry and technique of the present invention at the transmitter may have an advantage in that it includes an IIR like-equalization component (a group delay response) created by the impulse response of the communications channel. The leader and/or trailer taps of the equalization circuitry are adjustable "corrections" applied to the transmitted symbol. These "corrections" pulses are dispersed in the same manner that the transmitted symbol is dispersed by the impulse response of the channel. This effect may reduce or minimize the number of taps required to achieve a desired response and thereby improve or enhance the bandwidth of the transmitter output and the overall performance of the system.

[0119] There are many inventions described and illustrated herein. While certain embodiments, features, attributes and advantages of the inventions have been described and illustrated, it should be understood that many other, as well as different, embodiments, features, attributes and advantages of the present inventions that are apparent from the description, illustration and claims. As such, the embodiments, features, attributes and advantages of the inventions described and illustrated herein are not exhaustive and it should be understood that such other, as well as different, embodiments, features, attributes and advantages of the present inventions are within the scope of the present invention.

[0120] For example, the present invention may be implemented using more than three taps (See, for example, **FIG. 5B**). Indeed, some of the taps need not overlap but may be conventional type taps that have no overlap with the symbol or data signal and, as such, introduce no intentional intersymbol interference at the transmitter (see, for example, **FIGS. 12C, 12D, 12E** and **12F**). Thus, the present invention may be a combination of overlapping and non-overlapping taps.

[0121] Moreover, the present invention may be implemented using only leading tap(s) (see, for example, **FIG. 12A**), or trailing tap(s) (see, for example, **FIGS. 12B and 12C**), or as described above, using leading and trailing tap(s)(see, for example, **FIGS. 5A, 5B**, 6 and 11). Indeed, the present invention may be implemented using overlapping leading tap(s) or trailing tap(s) and non-overlapping leading and/or trailing tap(s) (see, for example, **FIGS. 12D, 12E** and **12F**). Other permutations of leading and trailing taps are suitable and are contemplated. As such, all permutations of leading and/or trailing taps having at least one tap that introduces intersymbol interference are intended to fall within the scope of the present invention.

[0122] In addition, the present invention may be implemented with taps having coefficients that are either positive or negative (i.e., digital C_{lead} and C_{trail} tap coefficients may be either positive or negative). As such, the taps may be

programmed or configured to add or subtract energy from the transmitted pulse. For example, in certain communications environments, there may be a reflection at a given frequency due to a known or predetermined impedance mismatches. In that circumstance, it may be advantageous to include a digital C_{trail} tap coefficient that is positive and, as such, adds energy to compensate or address the reflection (See for example, **FIG. 12G**). Similarly, in certain circumstances, it may be advantageous to include a digital C_{lead} tap coefficient that is positive and, as such, adds energy to compensate or address an unwanted response of the system (See for example, **FIG. 12H**).

[0123] Further, while in certain respects the present invention has been described in the context of an analog FIR filter having one or more taps positioned before and/or after the transmitted symbol, the present invention may be implemented using other types of equalization circuitry, for example, IIR filters.

[0124] In addition, while certain of the figures have illustrated the signals as pulse or square shaped signals, those illustrations should not be taken as limiting. Indeed, any shaped signals may be appropriate and the pulse shape of the equalization signal(s) is programmable or controllable so that the energy content of the equalization signals may be modified to provide the system additional flexibility in reducing, minimizing, mitigating or effectively eliminating pre-cursor and/or post-cursor intersymbol interference.

[0125] Further, it should be noted that communications channel **300** described herein may be, for example, constructed using one or more cables, wires, traces or the like, or may be part of a backplane, or may be a wireless communications medium through which the signal passes from transmitter **100** to receiver **200**. One skilled in the art will recognize that any such communications media, when used in conjunction with a corresponding transmitter/receiver pair appropriate for a particular medium, may be used to construct a communications channel in accordance with the present invention.

[0126] For example, other channels that may be implemented in the present invention include electronic, optical or wireless. Indeed, all types of channels of communication (i.e., communication channels), whether now known or later developed are intended to fall within the scope of the present invention.

[0127] In addition, it should be noted that other types of digital to analog converters may be implemented in the present invention. Indeed, any digital to analog converters, whether now know or later developed, may be implemented in the present invention to convert digital signals to an analog representation thereof.

[0128] It should be further noted that the term "circuit" may mean either a single component or a multiplicity of components, either active and/or passive, which are coupled together to provide or perform a desired function. The term "circuitry" may mean a circuit (whether integrated or otherwise), a group of such circuits, a processor(s), a processor(s) implementing software, or a combination of a circuit (whether integrated or otherwise), a group of such circuits, a processor(s) and/or a processor(s) implementing software. The term "signal" may mean a current or voltage signal whether in an analog or a digital form.

What is claimed is:

1. A system for providing data communication over a communications channel, wherein the communications channel includes a backplane, the system comprising:

a first transmitter coupled to the communications channel, the first transmitter includes equalization circuitry having at least one leading tap and at least one trailing tap wherein the at least one leading tap and the at least one trailing tap provide equalization signals that temporally overlap with a data signal and wherein the first transmitter outputs an equalized data signal; and

a first receiver, coupled to the communications channel, to receive the equalized data signal.

2. The system of claim 1 wherein the taps of the equalization circuitry include programmable coefficients.

3. The system of claim 1 wherein the taps of the equalization circuitry include programmable positioning.

4. The system of claim 1 wherein the taps of the equalization circuitry include programmable pulse durations.

5. The system of claim 1 wherein the taps of the equalization circuitry includes programmable coefficients and positioning.

6. The system of claim 1 wherein the taps of the equalization circuitry include programmable coefficients and wherein the system further includes a back channel for transmitting information which is representative of the coefficients of the taps of the equalization circuitry to the transmitter.

7. The system of claim 6 further including a second transmitter coupled to a second receiver via the communications channel, wherein the first receiver calculates information which is representative of the coefficients and wherein the second transmitter transmits the information which is representative of the coefficients to the second receiver.

8. The system of claim 7 wherein the first receiver periodically or intermittently calculates the information which is representative of the coefficients and wherein the second transmitter transmits the information which is representative of the coefficients to the second receiver.

9. The system of claim 1 wherein the first transmitter uses a PAM-4 communication technique.

10. The system of claim 1 wherein the taps of the equalization circuitry include programmable coefficients and wherein information which is representative of the programmable coefficients is provided to the system using an external interface.

11. A system for providing data communication over a communications channel, wherein the communications channel includes a backplane, the system comprising:

a first transmitter coupled to the communications channel, the transmitter includes equalization circuitry having a first trailing tap that provides an equalization signal that temporally overlaps with a data signal, wherein the transmitter generates an equalized data signal using the equalization signal and the data signal; and

a first receiver, coupled to the communications channel, to receive the equalized data signal.

12. The system of claim 11 wherein the first trailing tap of the equalization circuitry includes a programmable coefficient.

13. The system of claim 11 wherein the first trailing tap of the equalization circuitry includes a programmable position.

14. The system of claim 11 wherein the first trailing tap of the equalization circuitry includes a programmable pulse duration.

15. The system of claim 11 wherein the first trailing tap of the equalization circuitry includes a programmable coefficient and a programmable position.

16. The system of claim 11 wherein the first trailing tap of the equalization circuitry includes a programmable coefficient and wherein the system further includes a back channel for transmitting information which is representative of the coefficient of the tap of the equalization circuitry to the transmitter.

17. The system of claim 16 further including a second transmitter coupled to a second receiver via the communications channel, wherein the first receiver calculates information which is representative of the coefficient and wherein the second transmitter transmits the information which is representative of the coefficient to the second receiver.

18. The system of claim 17 wherein the first receiver periodically calculates the information which is representative of the coefficient and wherein the second transmitter transmits the information which is representative of the coefficient to the second receiver.

19. The system of claim 11 wherein the first trailing tap of the equalization circuitry includes programmable coefficient and wherein information which is representative of the programmable coefficient is provided to the system using an external interface.

20. The system of claim 11 wherein the equalization circuitry further includes a second trailing tap wherein the second trailing tap provides an equalization signal does not temporally overlap with the data signal, and wherein the transmitter generates an equalized data signal using the equalization signal provided by the first trailing tap, the equalization signal provided by the second trailing tap, and the data signal.

21. The system of claim 20 wherein the equalization circuitry further includes a leading tap wherein the leading tap provides an equalization signal that temporally overlaps with a data signal, and wherein the transmitter generates an equalized data signal using the equalization signal provided by the leading tap, the equalization signals provided by the first and second trailing taps, and the data signal.

22. A method for equalization of data signals that are transmitted over a communications channel, wherein the communications channel includes a backplane, the method comprising:

generating a first equalization signal using a leading tap wherein the first equalization signal temporally overlaps with a data signal;

generating a second equalization signal using a first trailing tap wherein the second equalization signal temporally overlaps with the data signal;

generating an equalized data signal using the first and second equalization signals with the data signal; and

transmitting the equalized data signal.

23. The method of claim 22 wherein generating a third equalization signal using a third trailing tap wherein the third equalization signal temporally overlaps with the data signal.

24. The method of claim 22 wherein generating a third equalization signal using a third trailing tap wherein the third equalization signal does not temporally overlap with the data signal.

25. The method of claim 22 further including controlling the second equalization signal by controlling the position of the trailing tap.

26. The method of claim 22 further including controlling the second equalization signal by controlling the coefficient of the trailing tap.

27. The method of claim 26 further including:

receiving the equalized data signal using a receiver

generating information which is representative of the coefficient of the trailing tap using the received equalized data signal;

providing the information which is representative of the coefficient of the trailing tap to the transmitter to control the coefficient of the trailing tap.

28. The method of claim 22 further including controlling the second equalization signal by controlling the pulse duration of the trailing tap.

29. The method of claim 22 further including controlling the first equalization signal by controlling the coefficient of the leading tap.

* * * * *

APPENDIX D

COPY OF WINSLAC SOFTWARE USER'S GUIDE (1999) ("WINSLAC")

WinSLAC
Software
User's Guide

1999

AMD

WinSLAC Software User's Guide

AMD

Table of Contents

Pub. # **22646** Rev: **A** Amendment: **/0**
Issue Date: **June 1999**

1.0 INTRODUCTION

The WinSLAC™ program is a software tool that aids in the design and development of telephone linecards and related voice band applications. It enables the user to design and generate coefficients for the programmable filters of the AMD SLAC™ family of devices and provides the user with predicted performance of system parameters.

The program models the SLAC device, the line conditions and associated linecard SLIC components. It calculates an optimum set of filter coefficients based on the overall system design conditions and generates the corresponding system responses for each of the programmable functions. It also calculates and plots predicted system responses for Two-Wire Return Loss (2WRL), Four-Wire Return Loss (4WRL), and Receive and Transmit frequency responses.

The WinSLAC program is implemented as a 16-bit Windows® application. It incorporates all of the functionality of the older AmSLAC2™, AmSLAC3™ and AmSLAC4™ programs, except for the DC feed computation capability of the AmSLAC3 program. The WinSLAC program also supports the Quad ISLAC and Dual ISLAC devices.

The WinSLAC program uses gain-phase parameters (G-Parameters) to describe the SLIC circuitry for input to the program. The G-Parameter arrays are typically produced by the program through SPICE simulation of the SLIC circuitry. They may also be entered manually using data obtained by lab measurements on a real SLIC circuit.

In order to generate the G-parameters, the WinSLAC program incorporates and uses an evaluation version of MicroSim Corporation's PSpice™ and Schematics™ programs to simulate the analog circuitry of the SLIC. Although the evaluation versions of these programs are sufficient for most designs, their limitations may impose certain restrictions on more complex designs. In such cases, the full production version of these programs may be purchased directly from MicroSim Corporation (now OrCad) and easily integrated into the WinSLAC program operation.

Note:
The MicroSim Corporation has been purchased by OrCad, Inc.; however, the WinSLAC software only supports versions 6.0 through 8.0 of the full production version of the MicroSim software available from Orcad, Inc. (http://www.orcad.com).

The WinSLAC software enables the user to design and generate an optimum set of coefficients for the programmable filters of the AMD SLAC family devices. The program will calculate coefficients for and generate the corresponding system responses for each of the programmable functions. Please note the following points:

1. It is recommended that you thoroughly read and understand this document before you attempt to install, load, and run the WinSLAC software program. There are figures to help the user visualize the screens that appear when running the program.
2. When running the WinSLAC software for the first time, it is suggested to run the program using the default parameters stored in the Examples directory which contains files that are installed with the software. AMD has included some sample output files which were created using the default parameters.
3. The WinSLAC program is designed to run on a "486" or above computer system running Microsoft Windows 3.1, Windows 95, or Windows NT®.
4. An important feature of the WinSLAC software is that it will predict the linecard's transmission performance (Two-Wire Return Loss, Four-Wire Return Loss, Transmit and Receive Attenuation Distortion) based on user inputs and the SLAC device's DSP Filter coefficients calculated by the program. This predicted performance assumes the perfect environment including zero component tolerances and no measurement errors. Since actual linecard testing will not be in a perfect environment, extreme care should be taken in choosing components and measuring these parameters if close correlation is to be established.

1.1 Distinctive Features

The WinSLAC program has these features and capabilities:

- Models the AMD DSLAC™, ASLAC™, QSLAC™, Quad ISLAC™, and Dual ISLAC devices.
- Calculates programmable coefficients for these devices for optimizing two-wire impedance, hybrid balance, and transmit and receive responses.
- Calculates and predicts linecard's transmission performance, such as:
  - Two-Wire Return Loss
  - Four-Wire Return Loss
  - Transmit and Receive attenuation distortion
  - Transmit and Receive path equalization
  - Two-wire Stability
  - DC Feed
- Provides visual edit of schematic and circuit files (using MicroSim Schematics™).
- Provides visual inspection of predicted system response graphs.
- Includes ability to adjust filter coefficients to modify system behavior.
- Compares system responses for two different designs.
- Includes ability to enter G-Parameter values directly into the program.
- Allows viewing, editing, and printing of files using user-specified editor/viewer.
- Copies system response graphs to paste into other Windows applications.
- Includes complete context-sensitive on-line help.

1.2 What the Program Does

The WinSLAC software is a tool that enables the user to design and generate an optimum set of coefficients for the programmable filters of the AMD SLAC family devices. The program contains an exact model of the devices, where the model is used for defining the complete transfer functions throughout program execution. Normally complete system responses include effects introduced by circuit components, including the SLIC circuitry. However, if the SLIC model is represented by simplistic ideal-gain blocks, the program output data will contain full response definition of the SLAC device only. This may be helpful to define certain system-level performance responses.

An important feature of the WinSLAC software is that it will predict the linecard's transmission performance (Two-Wire Return Loss, Four-Wire Return Loss, Transmit and Receive Attenuation Distortion) based on user inputs and the SLAC device's DSP Filter coefficients calculated by the program. This predicted performance assumes the perfect environment including zero component tolerances and no measurement errors. Since actual linecard testing will not be in a perfect environment, extreme care should be taken in choosing components and measuring these parameters if close correlation is to be established.

1.3 What the Program Does Not Do

The program has extensive capability for calculating an optimum set of DSP filter coefficient values based on the limits of system inputs. Due to the nature of interaction with programmable blocks within the SLAC device, these blocks which are most dependent upon this interaction with other blocks may have increased difficulty reaching optimum performance. An example of this is the closed loop impedance synthesis path of the ISN (Impedance Scaling Network), including AISN (Analog Impedance Scaling Network) or DISN (Digital Impedance Scaling Network), and Z-filter blocks with the forward gains within the SLIC to introduce a complex transfer function into

both the receive and transmit half-channel paths. This then requires compensation of this function by the individual receive and transmit path filters, in addition to their initial role of frequency response control. In cases of such interaction, the user must be able to make the determination of what limits must be imposed on the paths which have rippling effects upon overall system performance. For instance, if it is seen that there is difficulty upon meeting the necessary receive path frequency response performance, the problem can be reduced or even eliminated by manually forcing the Z-filter to relax its optimization of the two-wire synthesized impedance.

1.4 SLIC Model Representation

An evaluation version of MicroSim Corporation's PSpice and Schematics programs are included with the WinSLAC software and is used for simulating the analog circuitry of the SLIC in order to generate the G-parameters. The WinSLAC software supports versions 6.0 through 8.0 of the full production version.

1.5 Required Input

The program requires a complete representation of the entire analog portion of the circuit (i.e., the SLIC) which connects from the input and output of the SLAC device to the TIP/RING conductors of the line circuit. This is entered in schematic form, but may also be represented by a SPICE netlist or entered manually from lab measurements.

Additionally, there are numerous options that the user can specify for enabling or limiting individual programmable filter block calculations. By default, all programmable blocks are calculated automatically.

The program also requires a description for the market for which the linecard is being designed (system parameters). This includes the various impedances necessary for calculations as well as the templates the performance plots are required to meet.

Default system parameters and templates are provided for each device. These defaults can be changed by the user to fit their design. The system parameters and templates are also needed to obtain meaningful results.

1.6 Overview of WinSLAC Software

AMD's WinSLAC software is used to generate programmable coefficients for Advanced Micro Devices' SLAC™ family of programmable CODEC devices, and is also used as a development tool for designs using the AMD families of linecard devices. The software in this distribution is the WinSLAC software, and is used to generate programmable coefficients for Advanced Micro Devices' SLAC™ family devices.

1. ISN gain
2. Z-filter coefficients
3. B-filter coefficients
4. R-filter coefficients
6. X-filter coefficients
7. Transmit and receive gain coefficients
8. Predict (plot) two-wire return loss
9. Predict (plot) transhybrid balance
10. Predict (plot) receive path frequencies response
12. Predict (plot) transmit path frequencies response
13. Predict relative levels
14. Convert to a CSD value the default DCR1/2 settings
15. Convert to a CSD value the default LST setting

16. Predict (plot) two-wire stability bode response

The ISN, Z, and GZ filters are used to obtain impedance matching in the DSLAC device. The X and R filters help achieve frequency response requirements in the transmit and receive paths. The B filter helps meet transhybrid and/or adaptive balance specifications.

The ISN- or Z-, B-, R-, and X-filter blocks allow for individual setting options throughout the coefficient calculation. The program provides one of four choices for that selected block. These choices are:

- Calculate — Program will calculate coefficients (default selection).
- Disable Filter — Filter is cutoff and not used.
- Set — Coefficients (from manual calculation or previous run are manually entered).
- Read & Set — Loads value for the specified filter from an '*.ARF' file.

If selecting Transmit and Receive Gain, the program will prompt the user to provide the appropriate Gain Block settings. The choices for making these selections are:

- Manually Set AX and AR, Calculate GX and GR (default selection)
- Manually Set all Gain Blocks
- Calculate All Gain Blocks
- Disable (sets gains to 0db)

By using the Equalize option (selected from the *Templates* ➢ *Rcv/XMT menu*), the R- and X-filter coefficients can be generated to obtain the desired equalization in the Receive and Transmit paths, respectively. In this case, the receive and transmit path equalization response (amplitude and phase from 100 to 4000 Hz) must be provided as input to the program. The Equalize option may be necessary for applications requiring pre-emphasis, such as Channel Bank linecard designs.

The WinSLAC software has the following three options for SLIC usage:

- AMD's monolithic SLIC devices
- Transformer SLIC
- Any generic model SLIC (hybrid) including the transformer and AMD SLIC

Note:
The term SLIC as used with any SLAC device or in the WinSLAC software is defined as "Subscriber Line Interface Circuit," which identifies the entire circuit connecting the Four-wire analog port of the SLAC device to any externally-connected analog interface. The SLIC may be comprised of any active or passive circuitry, as well as a solid state (transformerless) line interface, which may itself be referred to as a "SLIC" by various manufacturers.

1.7 About The Manual

This manual has been rewritten and its organization is modified from that of earlier versions of the WinSLAC software. Much of the content from earlier versions is retained, but it has been restructured toward improvement of usability as well as incorporating program change topics. The manual is written with the assumption that the user is familiar with Microsoft® Windows®. All examples in this manual are for reference only unless otherwise stated.

2.0 GETTING STARTED

2.1 System Requirements

2.1.1 Hardware Requirements

- IBM-compatible PC with a minimum of 486DX processor
- Minimum of 8 MB of RAM
- Minimum of 10 MB of hard disk space

2.1.2 Operating System Requirements

- Windows version 3.1 or later
- Windows 95®, Windows 98® or Windows NT®

Note:
The PSpice and Schematics programs used by the WinSLAC software require "win32s" for proper operation on systems running Windows 3.1 or later.

If "win32s" is not already installed on your machine, you may obtain a copy from the Microsoft web page (http://support.microsoft.com/support/win32dev/faq/win32s/faq3474.asp).

2.2 Installation Instructions

The software will normally be distributed on three 1.4 MB diskettes or may have been supplied as a single compressed executable file. The installation must be performed from within Windows. To install the WinSLAC software from diskette, simply:

1. Insert disk 1 into the 3.5" floppy drive of your PC.
2. Use the "Run" command to execute the "setup.exe" program on disk 1. Alternatively, you may use Windows File Manager or Explorer to locate the "setup.exe" program on the floppy disk, then double click on the file name.
3. Follow the installation instructions on the screen.

If the software has been obtained through a network or from another computer containing the distribution file, it will exist as a single executable file. First run this executable from File Manager or Explorer, and it will extract the files that are contained on the distribution diskettes. Then, proceed with installation as in the previously mentioned steps 1-3, substituting the path location of the "setup.exe" program instead of disk 1.

2.3 Setting Program Parameters and Processing Delay Values

Delay values may be found in the following files located in the lib directory:

- Dslac.dly
- Aslac.dly
- Qslac.dly
- Qislac.dly
- Dislac.dly

Unless instructed to do so by AMD, these delay valves should not be changed.

3.0 USING THE WINSLAC PROGRAM

3.1 Basic Program Operation

The SLAC device can only be selected at the beginning of the program. Once a SLAC device is selected, it cannot be changed during a WinSLAC session. The reason for this will be explained later in the document. See Section 4.2, File Menu.

The program supports the DSLAC, ASLAC, QSLAC, Quad ISLAC and Dual ISLAC families of AMD's SLAC devices. It requires system parameters specification and definition of the SLIC as inputs. The WinSLAC program provides selection of computational options for individual filters of the SLAC device. It produces outputs of computed programmable coefficients for the device and prediction of system responses.

System parameters include design two-wire impedance, balance impedance, minimum performance templates for return loss, transmit and receive frequency response attenuation distortion templates, and design relative levels. Such parameters are normally specified by governing standards for the application of the design circuit.

The SLIC circuit (literally, everything between the external tip/ring connection and the analog input/output of the SLAC device) is represented in the form of four G-parameters, which identify gain and phase verses frequency. As these parameters are defined, any type of SLIC (AMD SLIC device, transformer, or any other combination) can be represented by these parameters. While it is possible to measure a typical circuit in the lab and record its measurements for entry into these G-parameter fields, circuit simulation using the included PSpice program will easily generate this data.

The System menu and Templates menu provide for system parameter entry. This is not required for SLIC definition or simulation, but is required before coefficient and response data can be computed. The SLIC menu is used to specify the SLIC circuit to be used and invokes the MicroSim Schematics™ and PSpice™ programs for circuit schematic entry and simulation. The SLAC menu allows individual filter and computational option selections. The Compute menu initiates program execution and computation. When the program is first started, the user is prompted for device (DSLAC, ASLAC, QSLAC, Quad ISLAC, or Dual ISLAC) selection. The program then continues until termination with support for that particular device.

For fundamental program operation, these basic steps can be followed:

1. Start the program from the program group which is created by the install program: *Start* ➢ *Programs* ➢ *WinSLAC* ➢ *WSLAC.EXE*.
2. Select the desired device (DSLAC, ASLAC, QSLAC, Quad ISLAC, or Dual ISLAC device) by clicking on the option button beside the device name.
3. Select the appropriate SLIC device: *SLIC* ➢ *Create Schematic* ➢ *[SLIC device]*.
4. A *Save As* dialog box will open, enter the desired name for the schematic file. This is in the form [SLIC device].sch. Clicking OK from this box invokes the Schematics program which shows a circuit of the SLIC.
5. Select circuit elements to modify and make circuit design changes as desired. See the appropriate SLIC datasheet for more information.
6. Create a Netlist: In MicroSim's Schematics, *Analysis* ➢ *Create Netlist*. The program will create a SPICE netlist file and supporting files that are used later in the simulation.
7. Exit the Schematics program. Click on Yes in the dialog box which appears asking to save the changes to this circuit.
8. Click on Yes in the next dialog box which appears asking to run PSpice. This box appears after the Schematics program has ended.

9. A dialog box now will appear asking for the circuit file to open. Click on the circuit name (this file will have the same base filename as the schematic when the netlist was created) to choose this circuit file name and then click on OK. This invokes the PSpice program which will then generate the G-parameter files.
10. Choose the desired system parameters for the design from the System and Templates menus.
    - Optionally, select the SLAC menu to choose options for individual filters of the SLAC device. Without selection, all filter data (except AX and AR gains) is computed by default.
    - Optionally, save the system parameters for future use: *File ➢ Save Parameter File.* It is not necessary to save the parameters unless it is desired to keep them; the computation uses the parameter data on the menu entry screens and does not require a saved file.
11. Select *Compute ➢ Filter Coefficients.* A dialog box will appear asking for the name of the G-parameter files and the name of the results (output of computation) files. A browse button allows easy selection without having to completely type the names.
12. From the *Select* dialog box, click on the G-parameter name and click OK to select it. After its path and name appears, click on the Compute button to invoke computation.
13. After computation is finished, a window of System Response Graphs will appear showing the computed responses of the system based on input data and calculated coefficient data.

The built-in on-line help provides most explanations and documentation of the program and also references supplied examples to guide the user through operation.

3.2 The Basic Functional Blocks of Linecard Design

Although a complete linecard contains many functions, those of concern relating to use of the WinSLAC software consist primarily of the analog "front end" line circuit and its transmission performance. The basic line circuit consists of an analog SLIC to provide the physical interface to the outside system, and the CODEC function for analog/digital conversion whose function is accomplished by the SLAC device. The WinSLAC program provides specifications of both of these major blocks, plus specification of the system parameters which represent the reference for performance conditions.

3.2.1 System Parameters

The System Parameters block provides specification of all transmission-related performance conditions including: impedance specifications, relative transmission levels, frequency responses, and performance requirements of two-wire and four-wire return loss.

3.2.2 SLIC Block

The SLIC block identifies all of the analog circuitry which exists between the SLAC device input and output connections and the tip/ring external connections. This includes not only the SLIC circuit itself, but all interconnecting elements, extra amplifiers or buffers, and protection components.

3.2.3 SLAC Device

The SLAC device block is the heart of the functional base of the program, as it is the program's purpose to calculate all of the SLAC device's programmable coefficient values. This menu selection provides choices of which programmable filters within the device should be included in the calculations. Options exist for each block to allow new computation, bypass computation, or manually specifying predetermined coefficient values, if desired.

3.3 Primary Computational and Execution Modes

The program's main purpose is to calculate coefficients for the programmable filter blocks of the SLAC device. There may be cases during the design process that certain filter values are not

needed to be calculated, or previous values are to be used instead. There are options for each programmable block to select the various modes for program execution.

3.3.1 Compute Coefficients Mode

This is the default condition for all programmable filter blocks (except for AX and AR) where the program determines all coefficient values for each of those blocks. The optimum set of coefficient values is derived and the resulting system level responses are generated and displayed by the program output.

3.3.2 Disable Mode

If it is not necessary to calculate the coefficient values for a particular filter block, the computational step for that selected filter block can be bypassed, allowing remaining executions to follow. The resulting system response is still displayed, but it is based on no compensation by the block which is disabled. By disabling all programmable blocks, no coefficient calculation will take place, but the system responses will still be generated. This is a good way to see the overall system performance without any interaction by the SLAC device itself, depending totally on the characteristics of the analog SLIC portion of the system.

3.3.3 Prediction Mode For Filter Calculations

A situation may arise where one or more of the SLAC device's filters requires manually entered coefficient values (including values from a previous program calculation) and use those values during calculation of the remaining filter blocks. This allows the user to slightly modify calculated values from the program in an effort to "fine tune" the filter responses. A filter which has its values manually set before program execution will not have new coefficients generated, but the program will still produce a predicted response based upon the values of the entered parameters.

An example of this is where the Z filter is attempting to synthesize a complex impedance that is very different from the Z_{SLIC} hard-wired in the circuit, and the R or X filters may not be able to perfectly compensate for the Z-filter's effects. By manually modifying Z-filter coefficients, which provide a more than adequate two-wire return loss, to a value that may produce a less than optimum but still acceptable value, enough frequency response distortion from the Z filter may be reduced to now allow the X or R filter to more effectively provide an overall flat response.

In evaluating effects of external variables, such as a variation of line or desired impedances, the WinSLAC software may be executed with all coefficients manually set to values from a previous program run. Since predicted results are still calculated and displayed, even without new coefficient calculations, this prediction mode of the program helps provide a worst-case description of overall system performance.

A wide range of decimal numbers may be manually entered for setting a coefficient, but only a discreet (although quite large) set of available codes are valid and within the ranges defined by the SLAC device's data sheet. The WinSLAC software will map the entered decimal number to the nearest available code. Therefore, this mapping can produce some degree of quantization error. For a range of approximately ±1.6, the available CSD codes are densely packed and quantization errors will be completely insignificant. Outside of this range, the available codes become slightly more widely spaced in certain regions and the quantization error may be larger. Although it still will be very minimal, the user should be aware of the differences. Since the decimal value for the coefficients that is displayed in the WinSLAC program output listing is generated from the available codes, this displayed number may not exactly equal a number that was manually entered before calculations.

3.4 Defining System Requirements and Parameters

The system requirements (referenced by the System menu) are used to specify the parameters which dictate performance and identify the external conditions which are connected to the telephone line circuit's tip/ring pair. These parameters are usually established by performance specifications from the appropriate regulatory agency for which the system will operate. These

system specifications include impedance, two- and four-wire return loss requirements, and receive and transmit path frequency response and signal levels and may be entered into the program into the following categories which are discussed in the next sections:

- Impedances
- Frequency Response
- Relative Levels
- Return Loss

3.4.1 Impedances

Three different impedances are used by the program:

1. Z_D: Desired (nominal) impedance is the two-wire impedance that is to be presented by the line circuit across tip/ring as seen from the two-wire line. This value is typically produced by choice of physical components which make up the analog portion of the line circuit. The impedance synthesis loop of the AISN/DISN and Z-filter block of the SLAC device can alter the effects of these physical components to create an impedance at the tip/ring terminals to meet the impedance value requirements. This Z_D value is used as one of the inputs to the program and may be specified by a ratio of s-domain polynomials or by magnitude and phase values over frequency.
2. Z_L: Line (Balance) impedance is the impedance presented by the two-wire line across the tip/ring of the line circuit that should produce the required transhybrid balance. A given value of Z_L acts as an external termination to the tip/ring terminals and will produce a specific amount of receive path reflected signal back into the transmit path. The B-filter within the SLAC device produces a matching reflected signal which cancels that from the analog path, thereby providing the line circuit's balance function. This Z_L value is also used as one of the inputs to the program and may be specified by a ratio of s-polynomials or by magnitude and phase values over frequency.
3. Z_T: Termination impedance is the impedance presented by the two-wire line to the tip/ring of the line circuit that is specified for external source or termination during gain, frequency response, and attenuation distortion measurements. The Z_T value is also one of the program inputs and may be left equal to the defined Z_D value or may alternately be specified as a ratio of s-polynomials.

3.4.2 Frequency Response

Frequency response requirements are used by the program and must be specified for each the receive (digital-to-analog) and transmit (analog-to-digital) paths. The entries for Attenuation Distortion provide an upper and lower boundary limit for the overall frequency response deviation about its nominal setting. Equalization entries provide the desired overall frequency response slope. These template values are entered into the program as dB values relative to a 1 kHz reference level.

3.4.3 Relative Levels

The "Relative Levels" are analog signal levels in dB across the tip/ring terminals which correspond to a reference 0dBm0 digital signal. The program uses these values to adjust the overall path gains in order to meet these specified analog signal levels. Separate entries are provided for both the receive and transmit paths.

3.4.4 Return Loss

User inputs for both two-wire return loss and four-wire return loss are provided. These are entered as template values in dB as a function of frequency and are used by the program to determine the extent of calculation necessary for computed coefficients in order to meet these system requirements.

3.5 How To Interpolate

The interpolate dialog may be activated from within any of the following dialogs by pressing the *interpolate* button:

- System Desired Impedance
- System Line Impedance
- System Return Loss Templates
- System Receive & Transmit Paths
- G-Parameter Values

Depending on how interpolate dialog is activated, the WinSLAC software attempts to guess at the initial values to use for the starting and ending frequencies, as well as the starting and ending interpolate values. The user, of course, may change any of these values, as desired. The *OK* button uses the displayed values to perform the interpolation, and fills in the appropriate table entries where the interpolated values are supposed to go. The *Cancel* button discards all changes, and does not perform any interpolation.

3.6 How to use the Read Filter Values Dialog

The "Read Filter Values" dialog may be used to read specific filter values from an ARF file (see Appendix A, WinSLAC Program File Overview). The filter values to be read may be specified by marking the appropriate check boxes.

This dialog may be activated from within any of the following dialogs by pressing the "*Read & Set*" button:

AISN or DISN & Z Filters

In this case, only AISN or DISN, ZIIR and/or ZFIR filter values may be read. Other options are disabled.

R & X Filters/Gain Blocks

In this case, only R and/or X filter values may be read. Other options are disabled.

B Filter & Adaptive Balance

In this case, only the B filter values may be read. Other options are disabled.

Global Settings

In this case, any of the filter values may be read.

3.7 How to Enter/Edit Values in Tables

Most of the user interface dialogs in the WinSLAC software provide a table for data entry. These tables have been designed to have a consistent usage.

Tables may be in one of two states, enabled or disabled. A disabled table, or a disabled cell within a given table, is shown in gray and may not be edited. When a table is disabled, it may be enabled through the appropriate radio button. This button is called "Set" in most filter dialogs. The exact operations are explained in the respective portion for each dialog.

An enabled table may be edited when it has the *input focus*. The table may get the input focus in any of the following methods:

- Single mouse click on a cell. In this case, the selected cell will be placed in overwrite mode, which is indicated by a border around the active cell.

- Double click on a cell. In this case, the selected cell will be placed in edit mode, which is indicated by the active cell shadowed.
- Tabbing through the controls on the dialog. In this case, when the table is tabbed to, the previously active cell will be placed in edit mode, which is indicated by the active cell shadowed.

When the active cell is in overwrite mode, the entire content of the cell will be overwritten with the values typed. On the other hand, when the active cell is in edit mode, the content of the cell may be edited. In this case, the left and right arrow keys may be used to move to the desired location within the cell to make the desired changes.

Tables in different dialogs have been designed to accept values that are valid for the particular purpose. The WinSLAC program beeps when the user attempts to enter a value outside the valid range for a given cell. The valid ranges for different parameters are specified in the respective sections of the manual.

The following table summarizes the methods that can be used to navigate through an enabled and active table.

Key	Action
Up Arrow	Moves active cell up one row
Down Arrow	Moves active cell down one row
Right Arrow	Moves active cell one column to the right, when the active cell is in overwrite mode.
	Moves the cursor right by one digit, when the active cell is in edit mode.
Left Arrow	Moves active cell left one column, when the active cell is in overwrite mode.
	Moves the cursor left by one digit, when the active cell is in edit mode.
Shift+Arrow Key	Extends selection in direction of arrow key
Page Up	Moves active cell one page up
Page Down	Moves active cell one page down
Ctrl+Page Up	Moves active cell one page left
Ctrl+Page Down	Moves active cell one page right
Home	Moves active cell to first cell in row
End	Moves active cell to last cell in row that contains data
Ctrl+Home	Moves active cell to first row, first column
Ctrl+End	Moves active cell to last row and column that contain data
Tab	Leaves the table, and sets the input focus to the next control on the dialog
Shift+Tab	Leaves the table, and sets the input focus to the previous control on the dialog
Shift+Space	Selects the current row
Ctrl+Space	Selects the current column
Shift+Ctrl+Space	Selects the entire table
Shift+DEL or Ctrl+X	Cuts the current selection or active cell's data to Clipboard
Shift+INS or Ctrl+V	Pastes Clipboard contents into active cell
Shift+INS or Ctrl+C	Copies the current selection or active cell's data to Clipboard

Note:
Some Windows settings, such as Large Fonts, may cause tables to display data poorly. To correct this, change your display to small fonts.

3.8 How to interact with the System Response Graphs dialog

The System Response Graphs dialog is activated either automatically after filter computations are performed, or as a result of selecting the "View-Graphs" option of the main menu. The behavior of the dialog is slightly different depending on the method of invocation, as described later in this section.

This dialog is designed primarily to display the predicted system responses in a graphical form, and to enable the user to make modifications to system and filter parameter values, and perform a re-computation. A comparison mechanism is provided to allow the user to compare two sets of system responses, as described in the following paragraphs.

When the dialog is first activated, it displays the following system response graphs all on one screen:

- Two-Wire Return Loss
- Four-Wire Return Loss
- Receive Attenuation Distortion
- Transmit Attenuation Distortion

If equalization is enabled for receive and transmit paths, the following two graphs are also shown.

- Receive Equalization
- Transmit Equalization

The plots shown in blue are the templates, and those in green are the predicted system responses. If the option to create the ".BOD" file is enabled, then the Bode plot is also available for viewing.

When a point on any of the graphs is double-clicked, the WinSLAC software displays the coordinates of the point in a bubble. The WinSLAC program also displays the actual coordinates of points that have been clipped and not the value shown in the graph.

3.9 Working With Examples

There are three examples provided with the WinSLAC software located in a directory called "examples". The purpose of these examples is to walk the user through creating and evaluating a linecard design. The first example uses a QSLAC device with an Am7920 SLIC model; the second uses a DSLAC device with an Am79R79 model; and the third uses the Quad ISLAC device with an Am79231 model and also with an Am79R241 model. Each example provides a readme file (readme.txt) that explains in detail the procedure to follow; those procedures are also found in Appendix B.

The first time these examples are run, the WinSLAC software users are encouraged to review them to familiarize themselves with the basic WinSLAC operations and to develop an understanding of the steps taken to create and evaluate a linecard design.

3.10 PCM and GCI file format

The ASLAC, QSLAC, Quad ISLAC and Dual ISLAC devices have both GCI and PCM options. One version supports the PCM data format and the other supports the GCI data format. The internal signal processing of both devices is identical, and coefficients generated by the WinSLAC program are also identical for each of these versions. However, due to the differences of command structures of these devices, the WinSLAC program must know what format is required for data output.

Select either PCM for the PCM devices, or select GCI for those devices.

3.11 DC Feed Files

3.11.1 ASLAC Device DC Feed

Note:

Presently, the WinSLAC software does not incorporate DC feed computational capability for the ASLAC device. That operation currently exists in AMD's AmSLAC3 Software, originally developed specifically for supporting the ASLIC/ASLAC device family. For this present version of WinSLAC software, it is necessary to use the DC feed computational capability of the AmSLAC3 software for determining all DC feed related parameters. Once those parameters are derived, they may be incorporated into the output data generated by the WinSLAC software by specifying the DC feed file names upon invoking "Compute."

The ASLIC/ASLAC devices are capable of regulating the DC voltage and current feeding a subscriber loop for any value of loop resistance. This function is referred to as DC feed and involves the programming of 7 ASLIC/ASLAC variables (VAPP, RFD, ILA, ILD, VAS, N2, and VOFF). The AmSLAC3 software aids in the setting of these variables to ensure that the amplifiers internal to the ASLIC do not clip the signals passing through them. This requires the user to enter many variables that describe the system so that the programmable variables determined by the program (VAS, N2, VOFF, and ILA. RFD is also calculated when the DC feed is in the constant current case.) are as accurate as possible. The AmSLAC3 program will aid in the determination of the optimum programmable variables to meet the user's specifications. However, occasionally the specifications will be too constrictive and the program will not find a solution for the given parameters.

Following is the suggested flow of operation for DC feed computation when using the WinSLAC program to develop final coefficients and output data for the ASLIC/ASLAC devices: Initial references are made to steps which must be followed for executing the AmSLAC3 program. After DC feed files have been generated, there is no need to continue with the AmSLAC3 program and the WinSLAC software can now be used for remaining file generation.

1. The AmSLAC3 software must first be used for all ASLAC DC feed computation. If it is not already loaded on your computer, please install it before proceeding. From a DOS prompt, invoke the AmSLAC3 software by typing AM3. The AmSLAC3 program will start normally.
2. Within the AmSLAC3 program, choose *SLIC ➢ AMD SLIC* type.
3. Select *SLIC ➢ Modify*, enter component values for the circuit and also the AC and DC subscriber line specifications. This will include setting some of the programmable variables as described in the AmSLAC3 User's Guide.
4. Also select *SLIC ➢ DC Feed* to initiate the DC feed variable optimization program. This will calculate the optimal VAS, N2, VOFF, ILA, and RFD (RFD is optimized only in the constant current case).
5. Examine the amplifier graph produced to ensure that the amplifiers are not clipping. Examine the loop V vs. I graph to ensure that it meets the specifications entered in the Modify menu.
6. At this point, exit the AmSLAC3 program. (It is not necessary to continue its execution or to select any other user options.) The data files needed for the WinSLAC program would have been generated. The WinSLAC software will copy the necessary data from those DC feed files and will automatically write the data to the ARF output file.

Because external components can effect the range of the DC feed feature, the user should calculate the DC feed variables prior to performing any of the SPICE simulations. Once the DC feed meets the user's requirements and all external components are set, the SPICE simulation and coefficient calculation can be performed.

3.11.2 Quad and Dual ISLAC Device DC Feed Parameters

The user has control over the parameters that are involved in the computation of the DC feed for the Quad and Dual ISLAC devices. These parameters are:

Rloop: This is the loop resistance to be fed. This line is plotted on the voltage verses current plot.

Ilim: Loop current limit. This sets the maximum current the ISLIC device will be able to source under a short circuit.

Rfd: This is the feed resistance in the constant resistance region, which is adjacent to the constant current region.

V1: This is the voltage at which the DC feed changes from constant current to constant resistance.

Vas: The anti-sat offset voltage. The anti-sat region is used to reduce the output voltage so that the amplifiers are not saturated by having an output voltage too close to the supply rails. The anti-sat voltage is the value of the battery being used minus Vas. The value needed will be application dependent and will need to be about 5V greater than any anticipated signal levels such as metering.

Rfdsat: This is the slope of the Dc feed curve while in the anti sat region. This region extend from the Y axis (open circuit voltage at zero current) to its intersection with the RFD line

PBAT: Value of the positive battery where applicable.

VBATH: 'High' negative battery, this is the most negative battery and is used for ringing and long loop DC feed

VBATL: 'Low' negative battery, this battery is the lower magnitude battery and is typically used for DC feed during the conversation part of the telephone call

RMGP: Thermal management resistor which connects from the TMP to pin to the TMS pin. This is used to supply some of the loop current to the load and remove some of the current from the ISLIC device to reduce ISLIC power dissipation

RMGL: Thermal management resistor which connects from the TMN pin to the VBATL pin. Since there is an internal switch between the VBATH and VBATL pins, the battery voltage selected for use appears at the Vbat2 pin. This is used to supply some of the loop current to the load and remove some of the loop current from the ISLIC device to reduce ISLIC power dissipation

STL_BAT: The VAB voltage at which the battery in use switches from VBL to VBH and from VBH to VBL. This voltage has 5 volts of hysteresis; for the transition from VBL to VBH it occurs when VAB reaches 2.5V above STL_BAT, and for the transition from VBH to VBL it occurs when VAB is 2.5V below STL_BAT.

THBAT: The high battery failure threshold. This sets the threshold level at which the device signals a battery failure for the VBATH

TLBAT: The high battery failure threshold. This sets the threshold level at which the device signals a battery failure for the VBATL

TPBAT: The high battery failure threshold. This sets the threshold level at which the device signals a battery failure for the VBATP

IFTD: sets the threshold for reporting of a DC fault current from ring to ground. This must exceed the ground key current to be able to distinguish a ground key from a DC fault, and a DC fault which is in to range of a ground key will be detected as a persistent ground key.

IFTA: sets the AC fault current threshold. This is measured in the ISLAC device by squaring and filtering the longitudinal current.

RTLL: Long loop ring trip threshold.

RTSL: Short loop ring trip threshold.

Tip- Ring Voltage vs. Current plot: This plot shows the variation of the tip ring voltage as the loop current changes. The open circuit voltage is at the point where the curve intersects the Y axis. Please see the Technical Reference for the ISLAC device being used for an explanation

Power vs. Resistance plot: This plots the power in the ISLIC device and the thermal management resistors as a function of the loop resistance.

Battery Current vs. Resistance plot: The battery current vs. loop resistance is plotted in the graph and gives an idea of required power supply currents.

ISLIC power vs. current: Power in the ISLIC device is plotted vs. the loop current to give an indication of ISLIC power dissipation using components chosen.

3.12 Out of Spec Points Description

The WinSLAC program will attempt to meet the system frequency response requirements described by the equalization and attenuation distortion templates. By altering the shape and limits of the templates, there is a considerable amount of control on how the filter responses are calculated: The filter calculation algorithms are iterative and during program execution will repeatedly calculate new coefficient values to attempt to meet the boundary limits. If boundary limits are wide, it is possible that the filters will make no or very little adjustment. If the boundary limit is too tight, it is possible to create a condition where convergence of the filter responses cannot be attained and the final response is limited by the program iteration control: Iteration of filter calculation stops when: 1) the boundary limits are met (per the allowed maximum number of "out of spec" points as defined in the WinSLAC program's Receive/Transmit template menu; or 2) when the responses of consecutive iterations no longer change.

If the iteration limit is reached by the program and the number of points outside the template boundaries has exceeded the maximum, as defined by this maximum out of spec limit, a message indicating this condition will be written into the output (*.ARF) file to warn of this condition. By changing this limit to a larger number, tighter constraint may be placed on the templates to attempt to force a desired response.

The value for "maximum out of spec points" can be modified from within *Templates ➢ Rcv/Xmt Paths* menu selection. The program contains an internal default value of 6 points, but the value may be changed to any number up through 40 points. (Setting to 40 allows the program to continue calculating as much as possible in "too tight" boundary conditions.)

3.13 Gain Tolerances Description

After initial coefficients have been computed, there may be a need to determine overall system performance degradation using those coefficients when a gain level changes within the SLIC circuit. The gain tolerance entry allows specifying such a gain variation.

This gain tolerance variation of up to +/-3 dB may be entered independently for both the receive and transmit directions.

3.14 PCM Encoding and Companding (A-Law, μ-Law) method

There are two choices for PCM encoding: A-law and μ-law. These are similar companding standards that are used to define the compression and expansion schemes of encoding the PCM data. The different companding schemes have slightly different magnitudes and this is taken into account in the WinSLAC program's optimization algorithms.

A-law is the digital compression and expansion standard used primarily in the European telephone network. μ-law is the standard used primarily in the North American and Japanese telephone networks.

3.15 Stability Compensation

The WinSLAC software automatically performs a stability analysis on the analog section of the SLIC and SLAC devices to check for any abnormal oscillations that could result due to instability. This stability checking is done due to the fact there is typically a closed loop gain from the SLAC

V_{TX} output, through the SLIC, back to the SLAC V_{RX} input, and through the AISN and Z-filter, back to the SLAC V_{TX} output.

Two conditions typically lead to instability: 1) A High Gain through the SLIC, which typically causes high frequency instability. 2) A High Gain at low frequency through the Z filter's IIR tap, where the denominator coefficient is close to unity. Of these two conditions, the WinSLAC program can control the ZIIR gain, and does so through reduction of the denominator coefficient value. Neither the program nor the SLAC device has any control over analog gains within the SLIC, and therefore no mechanism is available within this program to correct for such conditions. It is nevertheless checked, and appropriate warning messages are generated within the program's output. (The Criteria for stability is set within the WinSLAC program as a 3dB gain margin and a 10 degree phase margin.) An optional output file using the ".BOD" extension is created by the WinSLAC program which contains Bode plot stability data. This may be helpful in identifying stability conditions. If the option to create the Bode file is selected, the WinSLAC software allows the user to view the Bode plot.

If stability compensation is selected, the WinSLAC software will attempt to change the ZIIR coefficient if that is what is necessary to meet the stability margin criteria. If the compensation is not selected, the program will continue to execute, a warning of instability will be produced, and the .BOD file can then be used to determine the extent of instability which existed before any compensation is later chosen.

The results of .BOD file can be displayed by selecting the Bode radio button in view graphs. This will display magnitude and phase plots of the three loops which are tested for stability:

- **Internal loop** (green) — This is the loop that would result if Tip/Ring were shorted.
- **External loaded loop** (blue) — This is the loop representing normal linecard operation with Z_D as the load.
- **External unloaded loop** (red) — This is the loop that would result if Tip/Ring were open.

4.0 MENUS AND SCREENS

4.1 WinSLAC Operations

When the program first starts, it displays a dialog box from which the SLAC device may be selected. The devices currently supported are DSLAC, ASLAC, QSLAC, Quad ISLAC and Dual ISLAC device. The very first time the program runs, it comes up with the QSLAC as the default device. In subsequent runs, the WinSLAC software remembers the device last used and displays the dialog box with that device already selected.

Also, it should be noted that in the case of ASLAC device, the DC feed computations are not implemented in the WinSLAC software. Therefore, to use the WinSLAC software for the ASLAC device, the DC feed calculations must first be performed using the AmSLAC3 software. The resulting DC feed files may then be used with the WinSLAC software, as described in a later section. If DC feed computations are not needed, a default DC feed file will be suggested and used by the WinSLAC software.

The SLAC device can only be selected at the beginning of the program. Once a SLAC device is selected, it cannot be changed during a WinSLAC session. The reason for this will be explained later in the document.

4.1.1 Common Options

There are standard options found in most dialog boxes. These options perform the following functions unless otherwise noted.

OK

After all values are entered, this button exits from the dialog while saving the appropriate values for further use.

Cancel

This button exits from the dialog, discarding all changes that are made.

Help

This button displays WinSLAC help.

4.2 File Menu

The File menu provides read, save, save as, and view capabilities to the program. *Save* is limited to the saving of a parameter (.PAR) file which contains values for all user entry fields from the System menu and from the SLAC menu. *Read* capability allows reading in a previously saved parameter (.PAR) file to avoid having to re-enter all the menu values. The *View* capability allows viewing of all file types used by or created by the program.

4.2.1 View/Edit File

This option views, edits and/or prints a file. The toolbar button marked *View* (first button from left) is a shortcut to this menu option.

When this option is selected, a standard *Windows File Open* dialog is displayed, from which a file may be selected. The selected file is then opened in the editor specified in the file options dialog. The WinSLAC software is not able to execute more than one instance of a given editor.

If an attempt is made to open a new instance of an editor when an instance is already running, the behavior of the WinSLAC software depends on the behavior of the selected editor. If the editor allows more than one instance (such as Notepad or MSWord), the WinSLAC software attempts to detect multiple copies, and displays a message to that effect. If the editor does not

allow more than one instance, the WinSLAC software will just preload the selected file into the existing running instance.

4.2.2 Read Parameter File

This option reads a parameter file (see Appendix A, WinSLAC Program File Overview) previously saved by the WinSLAC software or any of the older AmSLAC programs. The toolbar button marked with the Windows file open icon (second button from left) is a shortcut to this menu option.

When this option is selected, a standard *Windows File Open* dialog is displayed, from which a parameter file may be selected. The parameters saved in the selected file are then read into the existing the WinSLAC software data structures.

If a PAR file was saved for a SLAC device other than the one in the current WinSLAC session, the WinSLAC software provides an option to read in only the system parameters, leaving the filter values/flags intact.

4.2.3 Save Parameter File

This option saves a previously opened parameter file.

If no parameter file is currently open, a *Save Parameter File As* dialog is displayed in which the user should specify a name under which the file will be saved.

4.2.4 Save Parameter File As

This option saves a parameter file with a new name. It displays a *Save Parameter File As* dialog in which the user should specify a name under which the file will be saved. The toolbar button marked with the Windows file save icon (third button from left) is a shortcut to this menu option.

4.2.5 Options

This option specifies what types of files should be generated.

Specify Files to Create

This section determines what types of files should be generated. Any combination of the following may be chosen:

- Desired Response File
- a Bode File (provides stability data)
- a Z_{IN} File (data representing the impedance looking into the SLIC)

Delete files

The PSpice .OUT files are normally deleted upon successful conclusion of the G-parameter PSpice calculations. If an error occurs during PSpice, this temporary directory will remain allowing the user to view the ".OUT" files to locate the error. Clicking on the check box until it is blank will keep these files from being deleted; a check in the box will cause the files to be deleted.

Editor to use

This option allows the user to specify the name of the editor to use to view/print files; the WinSLAC software uses the Windows notepad program by default. Use the Browse button to change the preferred editor.

Output Format

In the case of ASLAC, QSLAC, Quad ISLAC, and Dual ISLAC devices, the format of the output files may also be specified. Output files may be written in PCM format (the default) or GCI format. For more information see Appendix A.

4.2.6 Exit

This option exits the program.

4.3 System Menu

The System menu specifies the parameters that dictate required performance levels. This menu identifies the external conditions that are connected to the telephone line circuit's tip/ring pair. System menu parameters are established by performance specifications from the appropriate regulatory agency for which the system will operate. These system specifications include impedances, two- and four-wire return loss requirements, receive and transmit path frequency response, and signal levels. These parameters may be entered into the program into these categories:

Z_D: Desired (nominal) impedance is the two-wire impedance that is to be presented by the line circuit across tip/ring as seen from the two-wire line.

Z_L: Line (Balance) impedance is the impedance presented by the two-wire line across the tip/ring of the line circuit that should produce the required transhybrid balance.

Z_T: Termination impedance is the impedance presented by the two-wire line to the tip/ring of the line circuit that is specified for external source or termination during gain, frequency response, and attenuation distortion measurements.

Polynomial Format

Both real and complex impedances can be represented as a ratio of polynomials of powers of the s-domain Laplacian operator. The WinSLAC program accommodates such polynomials in s-domain with up to the fourth power of s in both the numerator and denominator. This allows the representation of complex impedance networks with up to four total inductors or capacitors in any series or parallel combination.

A complex and real impedance that is represented as a ratio of s-domain polynomials is expressed as:

$$Z(s) = \frac{(N0 + N1 \bullet s + N2 \bullet s\text{^}2 + N3 \bullet s\text{^}3 + N4 \bullet s\text{^}4)}{(D0 + D1 \bullet s + D2 \bullet s\text{^}2 + D3 \bullet s\text{^}3 + D4 \bullet s\text{^}4)}$$

The Nx and Dx terms represent the polynomial coefficients of the numerator and the denominator, respectively.

Magnitude/Phase Format

Both real and complex impedance values can be specified in terms of magnitude and phase as a function of frequency. Some telecom administrations (e.g., British Telecom) specify impedance parameters in this manner.

The magnitude and phase values are entered into a two-dimensional array. The array is 40 elements in length, corresponding to the 40 frequency points in 100 Hz steps from 100 Hz to 4000 Hz. Each array (frequency) entry contains a field for the magnitude, expressed in Ω, and phase, expressed in degrees or radians, if so selected.

Choice of degrees or radians is accomplished by selecting the appropriate radio button.

Four typical impedance networks that frequently occur in telephone line applications are shown with their representation as a polynomial in s-domain. The user can enter the numerator and

denominator values as decimal numbers calculated from the corresponding expressions in terms of the R and C resistor and capacitor values.

Figure 4-1 Impedance example 1: Single Resistor

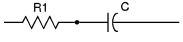
Where the impedance is real, not complex.

All s-domain terms of the polynomial are zero, and the impedance simplifies to: Z = R1

The corresponding WinSLAC program data entry is then:

	Numerator	Denominator
s0	R1	1
s1	0	0
s2	0	0
s3	0	0
s4	0	0

Figure 4-2 Impedance example 2: Complex impedance of R in series C



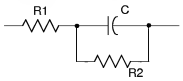
This impedance, written as a ratio of polynomials of the s-domain is:

$$Z = \frac{1 + (R1 \bullet C) \bullet s}{0 + (C) \bullet s}$$

The corresponding WinSLAC program data entry is then:

	Numerator	Denominator
s0	1	0
s1	R1*C	C
s2	0	0
s3	0	0
s4	0	0

Figure 4-3 **Impedance example 3: Complex impedance of R in series with a parallel RC**



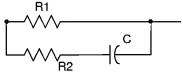
This impedance, written as a ratio of polynomials of the s-domain is:

$$Z = \frac{(R1 + R2) + (R1 \bullet R2 \bullet C) \bullet s}{1 + (R2 \bullet C) \bullet s}$$

The corresponding WinSLAC program data entry is then:

	Numerator	Denominator
s0	R1 + R2	1
s1	R1*R2*C	R2*C
s2	0	0
s3	0	0
s4	0	0

Figure 4-4 **Impedance example 4: Complex impedance of R in parallel with a series RC**



This impedance, written as a ratio of polynomials of s-domain is:

$$Z = \frac{(R1) + (R1 \bullet R2 \bullet C) \bullet s}{1 + [(R1 + R2) \bullet C] \bullet s}$$

The corresponding WinSLAC program data entry is then:

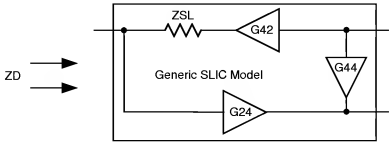
	Numerator	Denominator
s0	R1	1
s1	R1*R2*C	(R1+R2)*C
s2	0	0
s3	0	0
s4	0	0

4.3.1 Desired Impedance

Description

The Desired Impedance (Z_D) is the two-wire impedance, looking into and presented by the TIP and RING terminals. The WinSLAC program calculates programmable coefficients for the SLAC device (for AISN or DISN and for the Z-filter) in order to synthesize an impedance as close as possible to the value defined by Z_D. The corresponding two-wire return loss prediction in the program is presented as a measure of how closely the calculated impedance matches the value defined by Z_D.

Figure 4-5 Desired Impedance Z_D



The program accepts the Z_D data in the form of a ratio of polynomials in s-domain, or as an array of complex values given in Magnitude (in Ω) and Phase (in degrees). The desired impedance (Z_D) may also be copied to or copied from the line impedance for systems where these values are the same.

The WinSLAC program uses the entry mode selected by a set of toggle buttons in the Z_D dialog box. The buttons toggle between Magnitude/Phase entry and Polynomial entry forms.

When entering data as an array of complex values, the WinSLAC program uses phase in degrees as the default. Phase for impedances can be specified in radians. If radians are desired, select the appropriate radio button from within the Magnitude/Phase entry screen.

Several examples of actual desired (or nominal) input impedance requirements for selected countries are shown in Table 1.

Figure 4-6

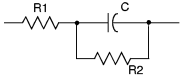


Table 1

Country	R1 (Ω)	R2 (Ω)	C (nF)
Australia	220	820	120
Austria	220	820	115
Belgium	150	830	72
China (PRC, old)	600	0	-
China (PRC, new)	200	680	100
France	215	1000	137
Finland	270	910	120
Germany	220	820	115
India	600	0	-
Italy	400	700	200
Japan	600	∞	1000
Korea	600	0	-

Country	R1 (Ω)	R2 (Ω)	C (nF)
Switzerland	220	820	115
United Kingdom	300	1000	220
U.S. & Canada	900	∞	2160

Note:

These specifications may not be correct or complete, or may not represent latest changes of some of the countries' requirements. These examples are provided for reference only.

Menu Option

The impedance menu option displays a dialog, in which the desired impedance parameter may be entered or modified.

The desired impedance can be entered in either s-domain polynomial format or as a tabular array of magnitude/phase format as a function of frequency. In addition, this parameter can be copied from the line impedance to save time and avoid duplication.

Form of Entry

The radio buttons in this group (*Polynomial* or *Mag/Phase)* may be used to specify the preferred form of parameter entry. When the dialog is initially displayed, the polynomial form is selected as default. However, if this option is changed, the WinSLAC software will remember the last setting during any given WinSLAC session.

Angle in

When the impedance parameters are entered in magnitude and phase form, the radio buttons in this group may be used to indicate the unit of measure (*Degrees* or *Radians*) used for the phase values.

When parameters are entered in polynomial form, the radio buttons do not have any meaning and therefore are disabled.

Copy from Z_L

This button may be used to copy line impedance parameter values to the desired impedance values. The setting of the Form of Entry decides whether the polynomial form or the magnitude and phase form of the line impedance parameters is copied.

Interpolate

When the impedance parameters are entered in magnitude and phase form, this button may be used to perform a linear interpolation of the magnitude and/or phase values.

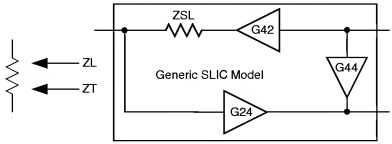
Pressing this button calls an interpolate dialog, in which the starting and ending frequencies, as well as the starting and ending values, may be specified. For details on the operation of the interpolate dialog, please see Section 3.5, How To Interpolate.

4.3.2 Line Impedance

Description

The Line Impedance (Z_L) is the impedance (for transhybrid balance) required to meet the transhybrid return loss or four-wire return loss specifications. It represents the value of the impedance that is placed across the design's tip/ring connection in order to define the optimum transhybrid balance condition.

Figure 4-7 Line Impedance for Balance and Test Impedance for Gain



ZL for four-wire balance tests

ZT for gain and frequency response

The WinSLAC program calculates programmable coefficients for the selected SLAC device (for the B-filter) to optimize the hybrid balance performance. The four-wire return loss prediction in the program is presented as a measure of how closely the calculated impedance matches the value defined by Z_L.

The balance impedance Z_L can be entered as a ratio of polynomials in the s-domain or as an array of values of magnitude (in Ω) and phase (in degrees) versus frequency. The Z_L value may be copied to or copied from the desired impedance for systems where these values are the same.

The WinSLAC program uses phase in degrees as the default entry for the array entry method. Phase for impedances can be specified in radians. If radians are desired, select the appropriate radio button from within the Magnitude/Phase entry screen.

Several examples of actual Line, or Balance Impedance requirements for selected countries are shown in Table 2:

Figure 4-8

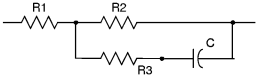


Table 2

Country	R1 (Ω)	R2 (Ω)	R3 (Ω)	C (nF)
Australia	600	0	∞	-
	220	820	0	120
Austria	220	1200	0	150
France	600	0	∞	-
Germany	220	820	0	115
Italy	400	700	0	200
Korea	600	0	0	-
Swiss	220	820	0	115
U. K.	370	620	0	310
U.S.A (Loaded)	0	100	1650	5

Country	R1 (Ω)	R2 (Ω)	R3 (Ω)	C (nF)
(Unloaded)	0	100	850	50
(Special)	900	∞	0	2160

Note:

These specifications may not be correct, complete, or represent the latest changes of some of the countries' requirements. These examples are provided for reference only.

Menu Option

This Line Impedance option displays a dialog, in which the line impedance parameter may be entered or modified.

Form of Entry

The radio buttons in this group (*Polynomial* or *Mag/Phase)* may be used to specify the preferred form of parameter entry. When the dialog is initially displayed, the polynomial form is selected as default. However, if this option is changed, the WinSLAC software will remember the last setting during any given WinSLAC session.

Angle in

When the impedance parameters are entered in magnitude and phase form, the radio buttons in this group may be used to indicate the unit of measure (*degrees* or *radians*) used for the phase values.

When parameters are entered in polynomial form, the radio buttons do not have any meaning, and therefore are disabled.

Copy from Z_D

This button may be used to copy desired impedance parameter values to the line impedance values. The setting of the Form of Entry decides whether the polynomial form or the magnitude and phase form of the desired impedance parameters is copied.

Interpolate

When the impedance parameters are entered in magnitude and phase form, this button may be used to perform a linear interpolation of the magnitude and/or phase values.

Pressing this button calls an interpolate dialog, in which the starting and ending frequencies, as well as the starting and ending values, may be specified. For details on the operation of the interpolate dialog, please see Section 3.5, How To Interpolate.

4.3.3 Termination Impedance

Description

The Termination Impedance (ZT) as defined within the WinSLAC program specifies the source and termination impedance of analog generation and measuring equipment that is connected to tip and ring of the linecard while making gain and frequency response measurements.

This ZT impedance is used in the WinSLAC program for calculating the selected SLAC device's programmable coefficients for the R-filter and X-filter and for the device's gain blocks.

In most cases this Z_T impedance is identical to the desired two-wire (Z_D) impedance, but there are known variations: e.g., the Bellcore LSSGR specification requires the two-wire input impedance to be 900 Ω + 2.16 uF; and the frequency response and gain is measured when the source impedance is 900 Ω.

The WinSLAC program default selection sets the termination impedance equal to the desired impedance, Z_D. This can be changed, if desired, and the termination impedance, Z_T value can be set. This impedance is entered only as a ratio of s-domain polynomials.

Menu Option

The Termination Impedance menu option displays a dialog, in which the termination impedance parameter may be entered or modified.

Use ZD for both

When this option is selected, the WinSLAC software uses desired impedance values for computations requiring Z_T. If the desired impedance is specified in polynomial form, the Z_D values are displayed in a table, so the user can see what values will be used. However, the table will be disabled to prevent changes to the values. If the desired impedance is specified in magnitude and phase format, a message to that effect is displayed to the user, and the values will be used for computations.

Specify ZT and use its value for both

When this option is selected, the WinSLAC software uses the utilizes specified Z_T values for both gain and frequency response calculations.

Specify ZT and use its value for frequency

Occasionally the impedance used for gain calculations is different than that used for frequency response. This option allows entry of the frequency response data in polynomial form while the gains are calculated using Z_D.

4.4 Templates Menu

4.4.1 Return Losses

Description

This option displays a dialog, in which the Return Losses parameters may be entered or modified.

Two-Wire Return Loss

The WinSLAC program calculates the programmable coefficients for the AISN/DISN and Z-filter blocks to synthesize a two-wire impedance to meet the minimum two-wire return loss defined by the specified two-wire return loss template. This two-wire return loss template is entered in dB against frequency.

The program will continue its iterative coefficient calculation for the filter coefficients in order to meet all of the requirements as described by the template. When either the template response is met by the predicted performance, or when the program has determined that no more iterations will improve the response beyond its current levels, computation then stops and those last calculated values will be used for the filter coefficients. This allows the user to control the filter coefficient values (and the return loss performance) by adjusting the template.

Note:

Since the AISN and Z-filter values (and therefore two-wire return loss) are the first ones calculated by the WinSLAC program, there may be an initial two-wire return loss performance that greatly exceeds the minimum system requirements. While this would normally be considered a benefit, there are other implications: The SLAC device's impedance synthesis loop (AISN & Z-filter) introduces an additional transfer function into both receive and transmit path. This addition, while providing a two-wire impedance that meets or exceeds actual requirements, also necessitates compensation (by the R filter and X filter) in receive and transmit paths. Also, that additional receive or transmit compensation can introduce a need for an even more complex B-filter transfer function for obtaining the needed four-wire

return loss. Therefore, in system cases where two-wire response appears to be very good, difficulty still arises in meeting other system responses. A relaxation of the desired performance, by reducing the two-wire template values, may help in meeting those other responses while still maintaining the minimum needs for the two-wire impedance.

Two-wire return loss (2WRL or TWRL) or simply "return loss" is a measure of the impedance match achieved between the two-wire TIP and RING leads and the nominal characteristic impedance (desired impedance) of the line. The PTT or the local telephone administration specifies the two-wire return loss requirements expressed in dB as a function of frequency, as a template. This template is entered into the WinSLAC program system parameters to define the minimum requirements at each frequency from 100 Hz to 4000 Hz.

When the impedance of the load (in this case, the line) matches the impedance of the source (in this case, the desired impedance) then the reflected signal from the load back to the source is minimized. Optimizing the desired impedance value within the WinSLAC program maximizes the 2WRL for the given design.

The CCITT requires the Return Loss to be > 16.5 dB at 300 Hz, 18 dB between 500 Hz and 2500 Hz, and 16.5 dB at 3400 Hz. Most countries follow the CCITT recommendations, while North America and British Telecom are notable exceptions. This CCITT recommendation for 2WRL is approximated as:

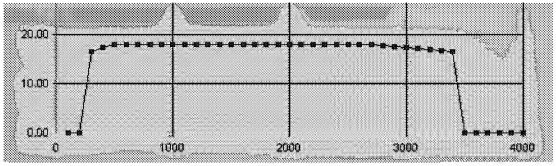
300 Hz: 16.5 dB

300 Hz to 499 Hz: rising linearly to 18.0 dB

500 Hz to 2500 Hz: 18.0 dB

2600 Hz to 3400 Hz: tapering linearly from 18 dB to 16.5 dB

Figure 4-9 Two-Wire Return Loss Template Graph



The LSSGR specifies the following requirement:

200 Hz to 499 Hz: 20 dB

500 Hz to 3400 Hz: 26 dB

The British Telecom requirements are:

200 Hz to 499 Hz: 18 dB

500 Hz to 2499 Hz: 20 dB

2500 Hz to 4000 Hz: 24 dB

Four-Wire Return Loss

The WinSLAC program calculates the programmable coefficients for the B-filter block to minimize reflections of the receive path signal which appear in the transmit path to meet the

minimum four-wire return loss defined by the specified four-wire return loss template. This Four-wire return loss template is entered in dB against frequency.

The program will continue its iterative coefficient calculation for the filter coefficients in order to meet all of the requirements as described by the template. When either the template response is met by the predicted performance or when the program has determined that no more iterations will improve the response beyond its current levels, computation then stops and those last calculated values will be used for the filter coefficients. This allows the user to control the filter coefficient values (and the return loss performance) by adjusting the template.

Four-wire return loss, transhybrid balance, or terminal balance return loss are all measurements of four-wire to four-wire echo cancellation achieved between the Receive (DRA) and Transmit (DXA) PCM ports of the SLAC device. This cancellation in the SLAC devices is achieved using the B-filter. These balance values represent a figure of merit for the B-filter performance. Expressed in dB, the four-wire return loss is numerically equal to transhybrid balance diminished by the sum of the insertion loss in each of transmit and receive paths.

The four-wire return loss specification and performance within the WinSLAC program is defined over the 100 Hz to 4000 Hz frequency range. Actual system performance specifications for various environments may be expressed in terms of a single numerical dB value of (SRL) singing return loss or (ERL) echo return loss. These single-number figures typically involve weighted summation of contributions of return loss over specified frequency ranges.

The CCITT requirement (approximate) is as follows:

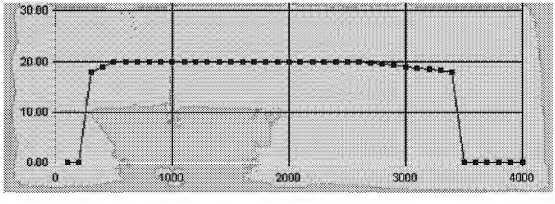
300 Hz: 18 dB

400 Hz: 19.0 dB

500 Hz to 2500 Hz: 20.0 dB

2600 Hz to 3400 Hz: tapering from 20 dB to 18 dB

Figure 4-10 Four-Wire Return Loss Template Graph



While many countries follow the CCITT recommendations, Bellcore suggests a more stringent requirement for the North American market.

The Bellcore requirements are (approximate):

200 Hz to 499 Hz: 20 dB

500 Hz to 2499 Hz: 25 dB

2500 Hz to 3400 Hz: 20 dB

Menu Options

Interpolate

This button can be used to perform a linear interpolation of the 2wrl or 4wrl values being entered.

Pressing this button calls an interpolate dialog, in which the starting and ending frequencies, as well as the starting and ending values, may be specified. For details on the operation of the interpolate dialog, please see Section 3.5, How To Interpolate.

Copy 2wrl to 4wrl

Pressing the Copy2wrl to 4wrl button copies the two-wire return loss values into the four-wire return loss values. These values will not be committed to memory until after the OK button is pressed.

Copy 4wrl to 2wrl

Pressing the Copy4wrl to 2wrl button copies the four-wire return loss values into the two-wire return loss values. These values will not be committed to memory until after the OK button is pressed.

4.4.2 Rcv/Xmt Paths

Description

- Frequency Response Requirement
- Relative Level (Lo/Li) of the Receive/Transmit Paths
- Equalization (to specify a required transmission line frequency response equalization characteristic)

The Z_T Termination Impedance is assumed to be present at the tip (a lead) and the ring (b lead) terminals for gain, attenuation distortion and frequency response calculations and measurement unless otherwise specified.

The receive path is the half-channel four-wire to two-wire (digital to analog) path describing the system response from the PCM digital interface to the terminated (by Z_T) tip/ring interface. The SLAC device's R-filter coefficients are calculated by the program in order to obtain a receive path frequency response to meet the boundaries described by receive path attenuation distortion templates.

The transmit path is the half-channel two-wire to four-wire (Analog to Digital) path describing the system response to the PCM digital interface from the terminated (by Z_T) tip/ring interface. The SLAC device's X-filter coefficients are calculated by the program in order to obtain a transmit path frequency response to meet the boundaries described by transmit path attenuation distortion templates.

4.4.2.1 Relative Levels Description

** Receive Relative Levels*

The Receive Relative Level (expressed in dBr) is the relative signal strength of a 1004 Hz analog signal appearing across tip and ring terminals when a 0 dBm digital signal is presented into the PCM input port of the SLAC device. This parameter is referred to as "Lo" in the CCITT Red and Blue books.

** Transmit Relative Levels*

The Transmit Relative Level (expressed in dBr) is the relative signal strength of a 1004 Hz analog signal sent into the TIP & RING terminals such that a 0 dBm digital signal appears on the PCM

output port of the SLAC device. This parameter is referred to as "Li" in the CCITT Red & Blue books.

The WinSLAC program uses the relative level setting to calculate the necessary gains (GX for the transmit path, GR for the receive path) within the SLAC device in order to set the system PATH gains required to meet the relative level specifications.

Note:
*This is **not** the linecard's path gain; it **is** a nominal terminated analog signal level referenced to a 0 dBm digital signal.*

The Z_T Termination Impedance is assumed to be present at the TIP (a lead) and the RING (b lead) terminals for gain, attenuation distortion and frequency response calculations and measurement unless otherwise selected.

4.4.2.2 *System Attenuation Distortion*

The lower and upper bounds of the allowable attenuation distortion as a function of frequency is entered in this table. There are 40 frequency points, from 100 Hz to 4000 Hz in 100 Hz steps. Rcv and Xmt each have two entry fields to establish the boundaries in dB of the receive absolute gain (attenuation distortion) limits, referenced to 1004 Hz. The WinSLAC software uses these values to optimize the attenuation distortion (frequency response) when performing a frequency response equalization. Note that a too stringent requirement in this respect may cause an error message to appear when the filter coefficients are being computed if the WinSLAC software is unable to generate the required response. (See additional comments within the attenuation distortion explanations.)

Attenuation distortion is the variation in gain with respect to frequency. Both upper and lower boundaries for the WinSLAC program are specified over the 100 Hz to 4000 Hz frequency range by data entered into the arrays.

These relative calculations are made with reference to the absolute system gain (or loss) at 1004 Hz. Normally, without equalization enabled (program default) the desired response is flat about a relative 0 dB reference. Therefore, the attenuation distortion will also represent the system frequency response. If equalization is enabled (which allows definition of a desired non-flat system frequency response) the attenuation distortion represents the system frequency response deviation from that defined system frequency response.

Filter Gain Normalization

The R and X filter coefficients are adjusted by the WinSLAC software to normalize the filter's response regarding the system gain.

Template Control Over Frequency Response Calculation

The WinSLAC program will attempt to meet the system frequency response requirements described by the equalization and attenuation distortion templates. By altering the shape and limits of the templates, there is a considerable amount of control on how the filter responses are calculated. The filter calculation algorithms are iterative and during program execution will repeatedly calculate new coefficient values to attempt to meet the boundary limits. If boundary limits are wide, it is possible that the filters will make no or very little adjustment. If the boundary limit is too tight, it is possible to create a condition where convergence of the filter responses cannot be attained and the final response is limited by the program iteration control. The Iteration of filter calculation stops when 1) the boundary limits are met (per the allowed maximum number of out of spec points as defined in the *RX/TX* dialog box or 2) when the responses of consecutive iterations no longer change.

The *RX/TX* dialog box contains a numerical value for maximum out of spec points which limits the iteration in cases where the iterations involving the R filter and the X filter do not converge. A message indicating this condition will be written into the output (*.ARF) file if this

condition should occur. (Setting to 40 allows the program to continue calculating as much as possible in too tight boundary conditions.)

Menu Option

The Receive/Transmit Paths option displays a dialog, in which system receive and transmit paths parameters may be entered. The table consists of seven columns that display the following parameters:

1. Frequency
2. Receive Path Attenuation Distortion (Lower dB)
3. Receive Path Attenuation Distortion (Upper dB)
4. Receive Path Equalization
5. Transmit Path Attenuation Distortion (Lower dB)
6. Transmit Path Attenuation Distortion (Upper dB)
7. Transmit Path Equalization

Equalization

The check boxes in this group allow the user to enable or disable the Equalization for both receive and transmit paths. By default, equalization for both paths is disabled.

Relative Levels

The Relative Levels for both receive and transmit path may be entered or modified here.

Number of Out of Spec Points

The maximum number of Out of spec points that are allowed during R and X filter calculations may be entered here. This value is set to 6 by default.

Copy to Rcv

This button may be used to copy all of the displayed values from transmit path attenuation distortion and equalization to the corresponding values in the receive path. Before the copy operation takes place, a dialog box appears that asks for confirmation from the user.

Copy to XMT

This button may be used to copy all of the displayed values from receive path attenuation distortion and equalization to the corresponding values in the transmit path. Before the copy operation takes place a dialog box appears that asks for confirmation from the user.

Interpolate

This button may be used to perform a linear interpolation for any of the columns in the table.

Pressing this button calls an interpolate dialog, in which the starting and ending frequencies, as well as the starting and ending values, may be specified. For details on the operation of the interpolate dialog, please see Section 3.5, How To Interpolate.

4.4.2.3 Equalization Description

Equalization must be enabled before any data can be entered in the equalization template array. Use the check boxes in the *RX/TX* dialog box to enable or disable equalization.

Enter the desired filter response in dB for each frequency. If the specification only provides dB values at wide frequency intervals it is possible to generate the extra values required by using the interpolate function.

The equalization function allows definition of an absolute frequency response. Initially, without equalization enabled (program default) the desired frequency response is flat about a relative 0 dB reference. Therefore, the attenuation distortion also represents the system frequency response. By enabling the equalization function and specifying the desired responses, separate frequency responses for both the receive and transmit paths may be independently selected. A number of performance specifications require such frequency responses.

The WinSLAC program uses input from the equalization specification plus the boundaries defined in the attenuation distortion specification to calculate R and X filter responses for receive, and transmit paths. The attenuation distortion templates establish an upper and lower boundary of the response, where the actual system response is described by the equalization templates. That is, the attenuation distortion boundaries represent the system frequency response DEVIATION from the system frequency response where the actual SYSTEM frequency response is established by the equalization table data. This enables the designer to produce a set of coefficients that will equalize the response of the entire transmission channel, considering the frequency response of the transmission line as well as the response of the linecard.

The equalization response is a response that is intended to be relative to the signal level at 1 kHz. Therefore, the template that is input to the program should pass through 0 dB at 1 kHz. If the template for the desired response is shifted from 0 dB at 1 kHz, there will be a corresponding gain shift in the system. Since the gain settings (GX and GR) are calculated after the frequency responses are calculated, and the gains are calculated at 1 kHz, the gain blocks will normalize the absolute signal levels as needed.

The transmit path is the half-channel two-wire to four-wire (analog to digital) path describing the system (combination of SLIC and SLAC device) response to the PCM digital interface from the terminated (by Z_T) tip/ring interface. The SLAC device's X-filter coefficients are calculated by the program in order to obtain a transmit path frequency response to meet the boundaries described by transmit path attenuation distortion and equalization templates.

Refer to additional notes describing attenuation distortion operation for further explanations regarding WinSLAC program operation for frequency response calculations.

4.5 SLIC Menu

4.5.1 Create Schematic

When a user clicks on this option, another pull-down menu with the allowable SLIC options will open. The SLIC options will be determined by the SLAC device chosen at the beginning of the WinSLAC session. To select the desired SLIC, move the mouse to the desired SLIC and click on it. This will open the schematic editor with the default schematic for the SLIC and SLAC chosen. See Section 6.1, MicroSim's Schematic Editor section for instructions on using the schematic editor. This schematic will be used, after user modification to generate a netlist, see Section 6.1.7, Create Netlist.

4.5.2 Modify Schematic

Click on this menu and a box will appear which will prompt for the selection of a schematic file to modify. This is used to change a schematic file which had been created at an earlier time. When the file is chosen (if the default directory is not the desired directory change the directory as needed in the folders or drives options) press the OK button and the schematic will be opened. Then modify the program and create a netlist as described in the Section 6.1, MicroSim's Schematic Editor and Section 6.1.7, Create Netlist sections.

4.5.3 PSpice

When this option is selected by clicking the mouse, a dialog box will open prompting to select a *.cir file. Select the desired file, changing drives or folders if necessary. Then press OK; this will start PSpice to create the G-parameters with the chosen file. The *.cir file is created by the schematics program. See the Section 6.1.7, Create Netlist section.

4.5.4 G-Parameter Entry

This option displays the dialog, from which the G-Parameter files may be read, viewed, modified and saved.

Show

The radio buttons in this group allow the user to select the set of G-Parameters to view. The selections are G24, G42, G44, or ZSL.

Read

This button displays the *Read the G-Parameters* dialog, from which previously generated or saved G-Parameters may be read.

Specify G-Parameters to Read

This dialog enables the user to select the set of G-Parameters to be read.

Specify Directory Path

If G-Parameters have been generated or recalled during the current session, the WinSLAC software displays the path and filename of the latest G-Parameters files used.

Write

This button displays the *Write the G-Parameters* dialog, from which one or more sets of the G-Parameters may be saved in user-specified files.

Specify G-Parameters to Write

This dialog enables the user to select the set of G-Parameters to be written to user-specified file(s).

Specify Directory Path

If G-Parameters have been generated or recalled during the current session, the WinSLAC software displays the path and filename of the latest G-Parameters files used.

Close

This button closes the dialog. If any of the G-Parameters have been modified, the user will be prompted to save or ignore the changes.

Interpolate

This button can be used to perform a linear interpolation of the magnitude and/or phase values. Pressing this button calls an interpolate dialog, in which the starting and ending frequencies, as well as the starting and ending values, may be specified. For details on the operation of the interpolate dialog, please see Section 3.5, How To Interpolate and Section 4.5.4, G-Parameter Entry.

This dialog enables the user to select the set of G-Parameters to be read. If G-Parameters have been generated, or recalled, during the current session, the WinSLAC software displays the path and filename of the latest G-Parameters files used.

4.6 SLAC Menu

The SLAC menu provides choices for coefficient calculation options for each of the programmable filter and gain blocks for the selected SLAC device. These filter and gain block options are AISN or DISN and Z Filters, R and X Filters, Gain Blocks, B Filter and Adaptive Balance, and Global Settings. Selecting one of these options opens individual menus for that particular block, where a choice can then be made to select the filter coefficient generation

options for that block. Each selection for a specific block acts independently from the other blocks during calculations.

These selections are:

Calculate

Enables the filter, allowing new coefficients to be calculated automatically for the filter. This is the program default.

Disable

No coefficients are calculated for the block if it is disabled.

Set

The coefficient values are manually entered by the user for the block.

Read & Set

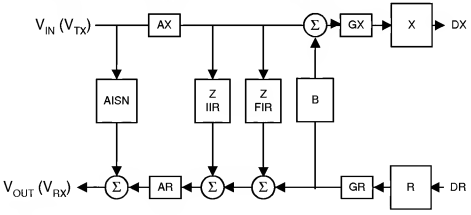
Load and sets a value from a saved file.

Choosing Disable effectively removes the filter from the functions of the circuit. The X and R filters can then be considered to have unity gain and the AISN/DISN, B, ZIIR, or ZFIR blocks can be considered as an open circuit.

Choosing Set Values for the ZIIR, ZFIR, B, R, and X filters will allow the user to enter the decimal values for the coefficients of that filter block. When this option is selected, the filter will be active (not disabled) within WinSLAC device's operation and the manually entered values will be used to define that filter's performance. This choice allows previously calculated coefficient values to be frozen for a particular filter while other filters are fine tuned and predicted performance is generated.

The Read & Set command selection will load values for the specified filters. Only the values for the selected filters will be changed. When a pathname has been specified a list of files under that pathname with the .ARF extension will be provided. Reading an output file (*.ARF) file will automatically copy all filter coefficients from that file.

Figure 4-11 Block Diagram of SLAC Devices Filter & Gain Blocks



Filters: AISN or DISN, ZIIR, ZFIR, B, AX, AR, GX, GR, X, and R

4.6.1 AISN & Z Filters

This option displays the dialog, from which the AISN (DISN in the case of the Quad ISLAC device) and Z filter parameters may be entered or modified.

AISN/DISN Filter

Note:
AISN is used for DSLAC, ASLAC, and QSLAC devices; DISN used for the Quad and Dual ISLAC devices.

This group of controls may be used to set the AISN/DISN flag and parameter value. By default, the AISN/DISN flag is set to calculate; and the box containing the AISN/DISN value is disabled. To set the AISN/DISN value, the Set Value radio button must first be selected. This enables the box in which the AISN/DISN value may be entered.

Calc

Enable filter, computer calculates new coefficients.

Set Value

Produce a menu screen of the device's 31 discrete gain steps the user may select.

Disable

Filter is an open circuit and removed from the remaining calculations.

ZFIR Filter

This group of controls may be used to set the ZFIR flag and parameter value. By default, the ZFIR flag is set to calculate, and the box containing the ZFIR value is disabled. To set the ZFIR parameters, the Set radio button must first be selected. This enables the box in which the ZFIR parameters may be entered.

Calc

Enable filter, computer calculates new coefficients.

Set

User manually enters 5 or 6 coefficient values for the filter depending on the device selected.

Disable

Filter open circuit and removed from the remaining calculations.

ZIIR Filter

This group of controls may be used to set the ZIIR flag and parameter value. By default, the ZIIR flag is set to calculate, and the box containing the ZIIR value is disabled. To set the ZIIR parameters, the Set radio button must first be selected. This enables the box in which the ZIIR parameters may be entered.

In the case of DSLAC device, the ZIIR Gz Settings group of controls may be used to set the Gz value, or let the WinSLAC software calculate it. The available settings in this group depend on the setting of the ZIIR flag. If the ZIIR flag is Calc, then Gz flag may be Calc or Set. If the ZIIR flag is Set, then Gz flag can only be Set. Finally, if the ZIIR flag is Disable, then the Gz flag is disabled, and the Gz value may not be set.

Calc

Enable filter, computer calculates new coefficients.

Set

User manually enters coefficient values for the filter.

Disable

Filter open circuit and removed from the remaining calculations.

The ZIIR Gz Settings group is enabled in the case of the DSLAC device only.

Calc

Enable filter, computer calculates new coefficients.

GZ Value

Produce a menu screen of the device's 8-bit shift option the user may select.

Disable

Filter open circuit and removed from the remaining calculations.

Note:
The ZIIR GZ Settings Option is used only for DSLAC applications. When ASLAC, QSLAC, Quad ISLAC or Dual ISLAC device is selected, (Z6) in ZIIR Options takes place of ZIIR GZ. ZIIR GZ shifts values up and down for calculations so there is no loss in significant digits.

Max ZIIR Setting

This group may be used to specify the maximum value of the ZIIR parameter or let the WinSLAC software use the default factory setting. Care must be taken when setting this value to something other than the default. To protect against accidental changes, when the Set button is pressed, a dialog appears that asks the user for confirmation.

Use Default

Sets maximum filter limit on recommended parameter of (0.998047), and the computer calculates new coefficients within this maximum limit.

Set Value

Manually entered maximum limit for the filter defined by user (not recommended).

Note:
ONLY users with great expertise in ZIIR calculation should attempt to adjust this critical value. A ZIIR max. limit set too high above the default will result in oscillation, and if set too low, it will limit the performance of the filter.

Read & Set

This option displays the *Read Filter Values* dialog, from which AISN/DISN, ZIIR and/or ZFIR filter values may be read from a given ARF file. For more information see: Section 7.2, SLAC Filter Analysis and Appendix A, WinSLAC Program File Overview.

4.6.2 R & X Filters/Gain Blocks

This option displays the dialog from which the R and X filters and gain block parameters may be entered or modified.

R Filter

This group of controls may be used to set the R filter flag and parameter values. By default, the flag is set to calculate, and the table containing the R filter value is disabled. To set the R filter values, the *Set* radio button must first be selected. This enables the table in which the values may be entered or modified.

Calc

Enables filter, computer calculates new coefficients.

Set

User manually enters seven coefficient values for the filter.

Disable

Filter set to unity gain and effectively removed from the remaining calculations.

X Filter

This group of controls may be used to set the X filter flag and parameter values. By default, the flag is set to calculate, and the table containing the X filter value is disabled. To set the X filter values, the *Set* radio button must first be selected. This enables the table in which the values may be entered or modified.

Calc

Enable filter, computer calculates new coefficients.

Set

User manually enters six coefficient values for the filter.

Disable

Filter set to unity gain, and effectively removed from the remaining calculations.

Gain Blocks (Gain Block Menu Options)

This group of controls may be used to set the AX, AR, GX and GR gain blocks flag and parameter values. By default, the flag is set to Set AX/AR, Calc GX/GR. The radio buttons may be used to change the flag settings, as desired. The possible choices are:

Set Ax/Ar, Calc Gx/Gr

Produces a menu screen with two gain settings for the analog gain and loss blocks, while the computer calculates new coefficients for the enabled digital gain/loss blocks.

This setting indicates that the AX and AR values will be set by the user to specified values, but the GX and GR values are to be calculated by the WinSLAC software. Selecting this option enables the AX and AR cells of the table to allow modifications and disables the GX and GR cells to prevent modifications.

Note:
Gain/loss blocks set at unity gain (0 dB) are equivalent to being disabled.

Set All

This setting indicates that AX, AR, GX and GR values will be set by the user to specific values. Selecting this option enables all of the table cells to allow modifications.

The user manually enters coefficient values for all four gain/loss blocks.

Calc All

This setting indicates that AX, AR, GX and GR values should be calculated by the WinSLAC software. Selecting this option disables all of the table cells to prevent modifications.

Computer calculates new coefficients for the four enabled gain/loss blocks.

Disable

This setting disables the gain blocks and all of the table cells to prevent modifications.

Set at unity gain, gain/loss blocks are removed from circuit during calculation.

Read & Set

This option displays the *Read Filter Values* dialog, from which R and X filter values as well as the gain block parameters, may be read from a given ARF file. For more information see Section 7.2, SLAC Filter Analysis and Appendix A, WinSLAC Program File Overview.

4.6.3 B Filter & Adaptive Balance

This option displays the dialog in which the B filter and Adaptive Balance values may be entered.

B Filter Values

This group of controls may be used to set the B filter flag and parameter values. By default, the flag is set to calculate. The table containing the BFIR filter values and the box containing the BIIR value are disabled by default as well. To set the B filter values, the Set radio button must first be selected. This enables the table and the box so that filter values may be entered/modified.

Calc

Enable filter, computer calculates new coefficients.

Set

User manually enters 8 or more coefficient values between (+/-3) for the filter. In the case of the Quad ISLAC device, the values are in Q1.15 format, which results in a range of –1 through +.999.

Disable

Filter open circuit and removed from the remaining calculations.

BIIR Value

User manually enters a coefficient value between (+/-4) for the filter.

Note:
B filter values are defaulted to (0).

Max BIIR Setting

This group may be used to specify the maximum value of the BIIR parameter or to let the WinSLAC software use the default factory setting. Care must be taken when setting this value to something other then the default. To protect against accidental changes, when the Set button is pressed, a dialog appears that asks the user for confirmation.

Default

Sets maximum filter limit on recommended parameter of (0.99000), and the computer calculates new coefficients within this maximum limit.

Set

Manually entered maximum limit for the filter defined by user (not recommended).

Note:
ONLY users with great expertise in BIIR calculation should attempt to adjust this critical value. A BIIR max. limit set too high above the default will result in oscillation and if set too low, it will limit the performance of the filter.

Adaptive Balance Parameters

This set of controls applies to all of the SLAC devices except the QSLAC device. They may be used to set flags and values relating to Adaptive Balance. Presently, the only function the WinSLAC software performs on these values is to convert them to the appropriate hex values in the ARF file. For more information, see the *Am79C02/03/031 DSLAC Devices Adaptive Balance Feature Application Note* (PID # 19693).

Digital Pre-Balance

This group of controls may be used to set the Digital Pre-Balance (DPB) flag and parameter value. By default, the flag is set to calculate, and the box containing the DPB value is

disabled. To set the DPB value, the Set radio button must first be selected. This enables the box in which the DPB value may be entered.

Calc
Enables filter, computer calculates new coefficients.

Set
User manually enters coefficient values for the filter.

Echo Path Gain

This group of controls may be used to set the Echo Path Gain (EPG) flag and parameter value. By default, the flag is set to calculate, and the box containing the EPG value is disabled. To set the EPG value, the Set radio button must first be selected. This enables the box in which the EPG value may be entered.

Calc
Enables filter, computer calculates new coefficients

Set
User manually enters coefficient values for the filter.

Error Level Threshold

This group of controls may be used to set the Error Level Threshold (ELT) flag and parameter value. By default, the flag is set to calculate and the box containing the ELT value is disabled. To set the ELT value, the Set radio button must first be selected. This enables the box in which the ELT value may be entered.

Calc
Enable filter, computer calculates new coefficients.

Set
User manually enters coefficient values for the filter.

Note:
Recommended default value for (ELT) is (-30.00).

DCR1

The Decorrelation Threshold (DCR1) value may be entered here. Manually set decorrelation threshold coefficients.

Note:
Recommended default value for (DCR1) is (0.250).

DCR2

The Decorrelation Threshold (DCR2) value may be entered here. Manually set decorrelation threshold coefficients.

Note:
Recommended default value for (DCR2) is (0.218750).

LST

The Low Signal Threshold (LST) value may be entered here. Manually set low level signal threshold coefficient.

Note:
Recommended default value for (LST) is (0.015625).

Read & Set

This option displays the *Read Filter Values* dialog, from which B filter values may be read from a given ARF file.

For more information see: Section 7.2, SLAC Filter Analysis and Appendix A, WinSLAC Program File Overview.

4.6.4 Global Settings

The Global Filter Settings option allows all programmable blocks to easily be Calculated, Set, or Disabled.

Global Filter Settings

Enables all filters and gain blocks for new coefficient calculations. The checkboxes in this group may be used as a shortcut to set all filter flags to Calculate, Set, or Disable. Only a maximum of one checkbox may be selected.

Set All Filters

Sets all filters to their default coefficient values. If Set All Filters is selected, then filter coefficients will not be calculated and the results generated by the compute function will be the results obtained as a result of using manually entered filter coefficients or the default filter coefficients obtained by a SLAC reset command. When all filters are designated to Read & Set, their coefficient registers will be loaded with all of their corresponding decimal values from another previously-generated output (ARF) file.

Note:
Default coefficient values are as follows: [AISN (DISN in the case of Quad ISLAC device), Z, and B filters; Ax, Ar, Gx, and Gr gain blocks] = (0); (X and R filters) = (1); For DSLAC device, (GZ setting) = (1); For ASLAC/QSLAC device, (Z6) = (1).

Disable All Filters

All Filters are essentially removed from coefficient calculations. Disable All Filters causes the selected filter to be cut off and not used in the resulting system design. The R and X filters become unity gain blocks allowing the signals to pass unaltered, and the AISN (DISN in the case of Quad and Dual ISLAC devices), Z, and B blocks are essentially open and effectively removed from the circuit. The output listing (*.ARF file) will still contain a predicted system performance for the filter(s) whose computation may be disabled and that listing will present CSD coefficients equal to the filter being in a disabled state.

Calculate All Filters

Enables all filters and gain blocks for new coefficient calculations.

Companding

The radio buttons in this group may be used to select the PCM encoding and Companding method. This selection does not apply to the DSLAC device; in the case of the DSLAC device, this group is disabled.

A-Law

European companding method selected.

μ-Law (Mu-Law)

North American and Japanese companding method selected.

Note:
See Section 3.14, PCM encoding and Companding method.

Gain Tolerances

The Gain tolerances may be entered in the edit boxes in this group. After initial coefficients have been computed, there may be a need to determine overall system performance degradation using those coefficients when a gain level changes within the SLIC circuit. The gain tolerance entry allows specifying such a gain variation.

This gain tolerance variation of up to +/-3 dB may be entered independently for both the receive and transmit directions.

Rcv Path

Manually entered gain variation of up to (+/-3 dB) for receive channel.

Xmt Path

Manually entered gain variation of up to (+/-3 dB) for transmit channel.

Stability Compensation

The radio buttons in this group may be used to specify whether Stability compensation should be enabled or disabled.

Disable

Stability is still analyzed, but no attempt is made by the program to compensate for any instabilities found.

Enable

This will induce a reduction in pole value to compensate for low frequency instability. See Section 3.15, Stability Compensation.

The software automatically performs a stability analysis on the analog section of the SLIC and SLAC device to check for any abnormal oscillations that could result due to instability. This stability checking is done due to the fact there is typically a closed loop gain from the SLAC device Vtx output, through the SLIC, back to the SLAC device Vrx input, and through the AISN and Z-filter, back to the SLAC Vtx output. Two conditions typically lead to instability: 1) A high gain through the SLIC, which typically causes high frequency instability. 2) A high gain at low frequency through the Z filter's IIR tap, where the denominator coefficient is close to unity.

Of these two conditions, the program can control the ZIIR gain, and does so through reduction of the denominator coefficient value. Neither the program nor the SLAC device has any control over analog gains within the SLIC, and therefore no mechanism is available within this program to correct for such conditions. It is nevertheless checked, and appropriate warning messages are generated within the output (.ARF) file. (The criteria for stability is set within the program as a 3 dB gain margin and a 10 degree phase margin.) An optional output file using the .BOD extension is created by the program which contains Bode plot stability data. This may be helpful in identifying stability conditions, and a plot is available when this in enabled.

If stability compensation is selected, the software will attempt to change the ZIIR coefficient if that is what is necessary to meet the stability margin criteria. If the compensation is not selected, the program will continue to execute, a warning of instability will be produced, and the .BOD file can then be used to determine the extent of instability which existed before any compensation is later chosen.

Read & Set

This option displays the *Read Filter Values* dialog (Section 3.6), from which specific filter values may be read from a given ARF file. See Appendix A, WinSLAC Program File Overview.

4.6.5 Display Filter Settings

This dialog is used for display purposes only, no editing is allowed. The dialog displays all the filter settings, the type of companding used, and the file format used. For information about the specific items in the display dialog, refer to the appropriate topics under Section 7.2, SLAC Filter Analysis.

4.7 Compute Menu

4.7.1 Filter Coefficients

This option displays the *Compute* dialog. In this dialog, the user can specify the names of the files required for computations, and the filename to use for the results of the computations, and then start the filter computations.

G-Parameters Files

The complete path and the base filename of the files containing the G-Parameter must be specified here. For example, if the G-Parameters are contained in the following files: g_param.g24, g_param.g42, g_param.g44, and g_param.zsl, and these files are located in the directory where they were stored when generated (for example: c:\winslac\test, then the entry in this box should read as: c:\winslac\test\g_param). The browse button may be used to aid with this file selection.

DC Feed Files

In the case of the ASLAC, Quad ISLAC and Dual ISLAC devices, DC feed files will be required for filter computations. The complete path and the base filename of the files containing the DC feed parameters must be specified here. The default DC feed files exist in the lib directory of the WinSLAC installation. If the DC Feed files are generated by the user, they can reside anywhere the user chooses.

The WinSLAC software currently supports DC feed calculations for the Quad and Dual ISLAC devices only. Therefore, to use the WinSLAC software for the ASLAC device, use either the default DC feed parameter files recommended by the WinSLAC software, or DC feed calculations must first be performed using the AmSLAC3 program, before filter computations can proceed. DC feed calculations do not apply to DSLAC and QSLAC devices.

Results Files

The complete path and the base filename of the result files must be specified here. The Browse button may be used to aid with this file selection. When the WinSLAC software completes the filter computations, it writes the results to a number of files with the specified base filename and different extensions. For example, if the specified base filename is results, then the WinSLAC software will generate the following files after filter computations are completed:

- results.arf
- results.hex
- results.prf

Furthermore, the following files may also be created, depending on the flag settings in the *File Options* dialog.

- results.bod
- results.drf
- results.zin

For more information see Appendix A, WinSLAC Program File Overview and Section 4.2, File Menu.

If during a given WinSLAC session, the *Compute* dialog is entered after Schematic files are created or modified, and/or PSpice simulations are performed, the WinSLAC software keeps track of the latest working directory and base filename, and assumes that the same directory and base filenames will be used. As such, it displays them in the appropriate boxes. The user, however, may change these selections.

Furthermore, when the Browse button selects the G-Parameters, the WinSLAC software assumes that the same directory and base filename will be used for the results, and displays the selection in the result box as well. The user, however, can change this selection.

Compute

When the Compute button is pressed, the WinSLAC software starts the filter computations. After the computations are successfully completed, the following system response graphs are displayed for the user to examine:

- Two-Wire Return Loss
- Four-Wire Return Loss
- Receive Attenuation Distortion
- Transmit Attenuation Distortion

If equalization is enabled for receive and transmit paths, the following two graphs are also shown:

- Receive Equalization
- Transmit Equalization

If the .BOD file was created, the stability Bode plot is also available.

If the response graphs are satisfactory, the user may press the Close button to close the *System Response Graphs* dialog, and return to the main window. However, the *System Response Graphs* dialog is designed to enable the user to make further modifications to the system and filter values, and start a new round of filter computations, if needed. For a detailed explanation of Section 3.8, How to interact with the *System Response Graphs* dialog, please see the respective section.

4.8 View-Graphs Menu

This option displays the *System Response Graphs* dialog, from which previously-generated system responses may be viewed. When the dialog is activated using this option, the following buttons are disabled:

- Globals
- Return Losses
- Rcv/Xmt Paths
- Aisn/Z Filter
- R & X/Gains
- B Fil/Adpt Bal
- Re-Compute

View Group

This group of buttons may be used to select which graph to view. When the *System Response Graphs* dialog is initially activated, it shows all of the graphs in one screen, and the *All* button is selected by default. However, any of the graphs may be viewed individually on a separate screen by selecting the appropriate radio button.

When only a single graph is displayed on a screen, the name of the file that contains the graph data is also displayed, along with the date and time of creation. Furthermore, the *Copy* button is enabled to allow the graph to be copied into the Window's clipboard.

If the response graphs were created with the equalization turned off for the receive and transmit paths, then the two buttons "EQRCV" and "EQXMT" do not apply, and therefore are disabled. If the option to create the ".BOD" file is selected, then the *Bode* button is enabled. Selecting this plot aides in finding instability problems.

Globals

This button displays the Globals dialog, from which the global settings may be modified. When the "Globals" dialog is activated from here, the "Read & Set" feature is disabled and hidden.

Return Losses

This button displays the Return Losses dialog, from which the two-wire and four-wire return loss parameters settings may be modified.

Rcv/Xmt Paths

This button displays Receive/Transmit Paths dialog, from which parameter settings may be modified.

AISN/Z Filter

This button displays the AISN or DISN and Z Filters dialog, from which parameter settings may be modified. When the "AISN & Z Filter" dialog is activated from here, the "Read & Set" feature is disabled and hidden.

R & X/Gains

This button displays the R & X Filters/Gain Blocks dialog, from which parameter settings may be modified. When the "R & X Filters/Gain Blocks" dialog is activated from here, the "Read & Set" feature is disabled and hidden.

B Fil/Adpt Bal

This button displays the B Filter and Adaptive Balance dialog, from which parameter settings may be modified. When the "B Filter and Adaptive Balance" dialog is activated from here, the "Read & Set" feature is disabled and hidden.

Compare

The compare feature allows the current predicted system response graphs to be compared with a previously generated set of graphs.

The "Compare" button operates like a toggle, or a two-way switch. In its OFF position, the button caption reads "Compare". In its ON position, it reads "Comparing", indicating that comparison is in progress.

When the Compare feature is turned ON, a Windows file open dialog appears that asks the user for the name of the file (with a PRF extension) containing the system responses to use for comparison. After the user selects the desired PRF file (see Appendix A, WinSLAC Program File Overview), the WinSLAC software reads the data from the selected file, and displays the system responses in **red**, alongside the current responses already displayed in **green**. The Compare button may be left in its ON position for as long as desired, in which case the PRF file selected previously will be used for comparison purposes. If comparison with a new set of responses is desired, the compare feature must first be turned OFF, and then ON again. The user may then select a different file to compare with.

When the Compare feature is turned OFF, the comparison ends, and the system responses displayed in red disappear.

Zoom In

The "Zoom" feature allows the user to zoom in to a specific portion of any of the graphs to view it more closely. The "Zoom" button operates like a toggle, or a two-way switch. In its OFF position, the button caption reads "Zoom In". In its ON position, it reads "Zoom Out", indicating that zoom feature is already enabled.

When the zoom feature is enabled, the user may draw a rectangle around the graph area to be enlarged. Please note that only the area enclosed by the rectangle will be magnified. Therefore, it is recommended that one or both axis can be included in the zoom region, so that the values be displayed.

Print

Pressing the "Print" button sends each of the graphs currently displaying to the default printer. Each graph is sent to the printer separately.

Copy

The "Copy" feature allows the graphs to be copied into the Windows clipboard, so that they may be used in other Windows applications. When only a single graph is being displayed, clicking on the "Copy" button automatically copies the graph onto the clipboard.

This feature is disabled when multiple graphs are displayed.

Receive Relative Level

Shows the computed value of the Receive relative level. When comparing graphs, a second box appears. This box shows the Receive relative level value corresponding to the graph you are comparing with.

Transmit Relative Level

Shows the computed value of the Transmit relative level. When comparing graphs, a second box appears. This box shows the Transmit relative level value corresponding to the graph you are comparing with.

Re-Compute

Pressing this button starts a new round of filter computations. It operates the same as the Compute menu item from the main menu of the WinSLAC software.

Close

This button closes the "System Response Graphs" dialog, and returns to the main window of the WinSLAC software.

4.9 Help Menu

This selection displays the topics available in WinSLAC's help file.

4.9.2 Using Help

This selection displays help on how to use the help system.

4.9.3 About WinSLAC

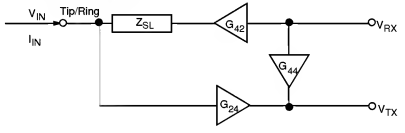
This selection shows the version and copyright info about the WinSLAC software program.

5.0 SLIC MODELS AND REFERENCE

5.1 SLIC Functions and G-Parameter Representations

A SLIC (Subscriber Line Interface Circuit) consists of a circuit which connects an analog telephony four-wire port (separated transmit and receive paths) to an analog telephony two-wire port (the two-wire connection which connects to the telephone and which carries combined receive and transmit signals). The SLIC can be implemented by an AMD SLIC device with surrounding components or a combination of active and passive gain and impedance blocks. The SLIC may also include a precancellation network, used to perform some transmit signal cancellation from the receive signal appearing on a two-wire line interface. Any SLIC can be modeled in the form shown in Figure 5-1, consisting of three gain blocks and one impedance block. These four functional blocks are referred to as G-Parameters.

Figure 5-1 The G-Parameter SLIC Model



For this G-Parameter representation, the input impedance at the four-wire input (V_{RX}) is infinity, and the four-wire output (V_{TX}) impedance is zero. The SLICs' operation can be represented by the two equations:

$$V_{IN} = Z_{SL} * I_{IN} + G_{42} * V_{RX}$$

$$V_{TX} = G_{24} * V_{IN} + G_{44} * V_{RX}$$

$$Z_{SL} = \left.\frac{V_{IN}}{I_{IN}}\right|_{V_{RX}=0} \qquad G_{24} = \left.\frac{V_{TX}}{V_{IN}}\right|_{V_{RX}=0}$$

$$G_{42} = \left.\frac{V_{IN}}{V_{RX}}\right|_{I_{IN}=0} \qquad G_{44} = \left.\frac{V_{TX}}{V_{RX}}\right|_{V_{IN}=0}$$

The G-Parameters can be measured on an actual circuit using an accurate gain/phase impedance analyzer, similar to the Hewlett Packard model 4192 instrument. G_{24} is the gain from the two-wire port Tip/Ring, which is both an input and an output, to the four-wire output of the SLIC (VTX); G_{42} is the gain from the four-wire input (V_{RX}) to the two-wire port tip/ring; G_{44} is the gain from the four-wire input (V_{RX}) back to the four-wire output (V_{TX}); and Z_{SL} is the output impedance at the two-wire port. The sources V_{IN} and I_{IN} are used as test sources depending on what needs to be calculated. When calculating the value of V_{IN}, I_{IN} is used as the source and V_{IN} calculated. When calculating V_{TX}, V_{IN} is a source and I_{IN} is zero. For an ideal SLIC, G_{24} and G_{42} will have constant gain and phase and are typically unity gain and zero phase, and G_{44} is equal to zero. Z_{SL} would then equal the actual desired source or termination impedance for the two-wire interface. In an actual circuit, however, these G-Parameter values will vary in both magnitude and phase, representing real and complex circuit functions.

The G-parameter representation shown is a single output model. The model for the SLIC has a single-ended output also, and the conversion from a double output to a single output will be discussed briefly in Section 5.5, Component Selection. The node labeled tip/ring is the single ended equivalent of the tip and ring pins.

Simplified models for measurement or definition of the G-Parameters are shown in Figure 5-2. This figure shows a SLIC circuit connected in each of the four different arrangements that provide for measuring the G-Parameters.

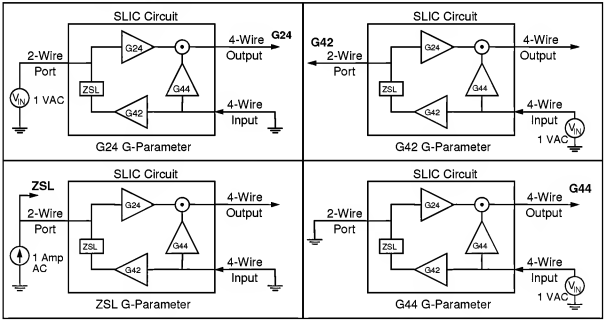
The G_{24} G-Parameter is defined as the gain from the two-wire interface to the four-wire output. This is measured by connecting a 1 volt AC voltage source to the SLIC's two-wire interface and measuring G_{24} at the four-wire output. This configuration is shown in Figure 5-2 in the top left quadrant of the figure. The four-wire input is grounded for this measurement.

The G_{42} G-Parameter is defined as the gain from the four-wire input of the two-wire interface. This is measured by connecting a 1 volt AC source to the SLIC's four-wire input and measuring G_{42} across the two-wire port. This is shown in Figure 5-2 in the top right quadrant of the figure.

The Z_{SL} G-Parameter is defined as the input impedance looking into the SLIC's two-wire interface. This value can be measured by connecting a 1 Amp AC current generator at the two-wire interface, where the voltage magnitude and phase measured at that point equals the value of Z_{SL}. The four-wire input is grounded for this measurement. This is shown in Figure 5-2 in the bottom left quadrant of the figure.

The G_{44} G-Parameter is defined as the gain from the SLIC's four-wire input to its four-wire output. This is measured by connecting a 1 volt AC voltage source to the four-wire input and measuring G_{44} at the four-wire output. This is shown in Figure 5-2 in the bottom right quadrant of the figure.The two-wire interface is grounded for this measurement.

Figure 5-2 G-Parameter Model SLIC Representations



The actual values of these G-Parameters can be modified by externally applied gains at the four-wire port, which is the means that is used by the SLAC devices' AISN- and Z-filter blocks to meet various impedance requirements at the two-wire port for fixed values of these G-Parameters.

5.2 SLIC Model for The WinSLAC Program

AMD solid state SLIC models are defined within the WinSLAC program. The schematic entry program allows entering the desired values for the components of the model and for changing the schematics of the default circuit. The appropriate SPICE circuit is built by the schematic entry program when the Create Netlist option is selected from the Analysis menu in the schematic entry program. Using this SPICE netlist for the SLIC circuit, SPICE input files (one for each of the

G-Parameters) are then built, using the measurement technique described in the preceding section.

5.3 AMD SLIC Models

The WinSLAC software includes simplified SPICE circuit models for a number of AMD's SLIC devices. These individual files are installed in the WINSLAC\LIB directory. These models do not represent all of the functional aspects of each of the SLIC devices but do represent the AC transmission characteristics of each device as well as the DC characteristics necessary to define the AC performance under limited DC loop conditions. The PSpice program supplied with the WinSLAC software is an evaluation version and is limited as to the number of circuit elements and nodes permitted. The presently supplied models are near that limit. More detailed models are not feasible using this version of PSpice. To model the AC system performance and calculate SLAC coefficients, these models are sufficient.

The SLIC.LIB file in WinSLAC\LIB directory must contain that device's model name. If any models are to be used which are not in this original distribution they must be copied into the WinSLAC\LIB directory and the SLIC.LIB file must be edited to add the new name. The SLIC.IND file (also in the \LIB directory) should be deleted whenever the SLIC.LIB file is modified; it is an index file created the first time the PSpice program runs and is thereafter used as a pointer. After adding a SLIC name entry into the SLIC.LIB file, deleting the SLIC.IND file will cause PSpice to create a new SLIC.IND file to incorporate the change to the SLIC library. To use a new SLIC that does not appear in the SLIC ➢ Create Schematic menu after modifying the SLIC.LIB file, choose any SLIC in the Create Schematic menu item. Delete the SLIC that appears in the schematic in PSpice and replace it with the new SLIC by using the Draw ➢ Get New Part menu selection. At the Add dialog box, select Browse, then select the SLIC.SLB or SLIC2.SLB library, and select the new SLIC that was added from the menu and place it on the schematic in place of the original SLIC.

This release (v1.0) of the WinSLAC software contains circuit models for these SLIC devices:

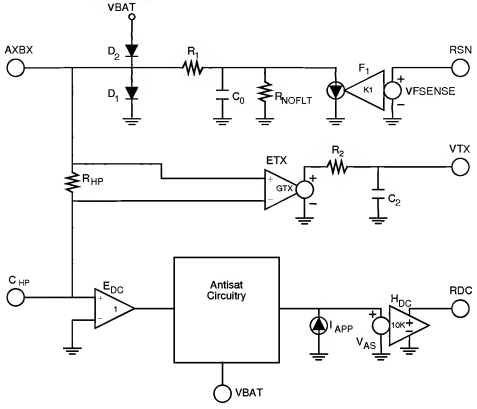
- Am7920
- Am7922
- Am7942
- Am7943
- Am7944
- Am7945
- Am7946
- Am7947
- Am7948x
- Am79484
- Am79489
- Am7949
- Am7950
- Am7953x
- Am7953xi (Model is copied from the Am7953x model)
- Am7957x
- Am79512 (Model is copied from the Am7953x model)
- Am79M53x
- Am79M57x
- Am79R79
- Am79R70
- Am79213
- Am79231
- Am79R241
- Am79R251
- Am79R100
- Am79R101
- Transformer
- ASLIC

Contact AMD for updates to models as they become available, or for model usage for devices which may not be listed.

Most of the models are similar to the circuit shown in Figure 5-3 with the main difference being the antisat circuitry. Each of the models has been correlated with lab results for accuracy. The current amplifier gain (K_I) and transmit voltage amplifier gain (G_{TX}) may differ from model to

model and are specified in the databook. The schematic entry program generates node numbers as appropriate.

Figure 5-3 Internal Circuitry for AMD SLICs



5.4 The AMD SLIC SPICE Model

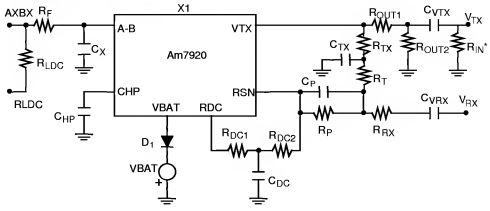
The AMD SLIC circuit model, shown in Figure 5-4, is the default circuit that the schematic entry program uses to build the SPICE schematic file, FILENAME.SCH. This shows the use of an Am7920 in the circuit configuration to connect to a QSLAC device.

There are two important points to keep in mind when entering the schematic:

- **This configuration will vary among the SLAC families**. The default configuration will be different for a QSLAC, DSLAC, Quad ISLAC, Dual ISLAC or ASLAC devices and the schematic shown is representative for the QSLAC device only.
- **The default values for the external components related to setting the DC loop and transmission characteristics will most likely be incorrect**. There is only one set of default components for all SLIC's; the chance that the component values are correct for any given application are slim. Always enter the component values when creating a new schematic.

The schematic may be modified as desired, but the labeled input and output nodes (VTX, VRX, RLDC, AXBX) must keep their names to ensure correct connectivity. The resistor R_{LDC} is the DC load of the line circuit to set the DC operating point of the SLIC. When SPICE simulation is performed which allows for a DC bias point to be established representative of DC loop conditions, an inductor (inductor L_L) is connected in series with RLDC to form a DC holding circuit which does not affect the AC performance.

Figure 5-4 The AMD SLIC SPICE Model



5.5 Component Selection

The single ended model used requires adjustment to four components for proper simulation. The components are C_{AX}, C_{BX}, R_{FA}, and R_{FB}. Conversion from double ended to single ended is fairly straightforward. The two nodes A and B become a single node AXBX referenced to ground which has a voltage twice that of each of the individual nodes. The C_{AX} and C_{BX} are combined into a single capacitor to ground with a value of half what each one was. The fuse resistors are summed into a single R_F that is twice the value of the individual fuse resistors that would appear on the final circuit.

The components on the schematic are included as a default configuration; many of the component values are application dependent, and some are dependent on the SLIC type.

R_{LDC} is used as part of the simulation to set the DC loop, but it would not be part of an application. The telephone and line impedance of the specific application would provide the DC loop.

C_X is a single capacitor of half the C_{AX} value which substitutes for the C_{AX} and C_{BX} capacitors that connect from tip to ground and ring to ground. These provide EMI protection in a typical application. R_F is twice the fuse resistors and provides fault and over-voltage protection in conjunction with a protection device not shown, since it does not have any effect on the AC signal path.

The data sheet for a specific SLIC will have the necessary equations to calculate R_{DC1}, R_{DC2}, and C_{DC} to set the DC loop current limit and the DC loop time constant.

R_P and C_P are provided to allow the use of complex networks, and are typically set to very small values so they have negligible effect on typical applications

The data sheet for a specific SLIC will have a C_{TX} value along with R_T and R_{TX}, generally shown in Note 1 in the data sheet. This sets a time constant that provides a delay compensation for the circuit. If R_T and R_{TX} are not the same as shown in the data sheet, recalculate C_{TX} to get the same time constant as calculated with the data sheet values.

R_T, R_{TX}, and R_{RX} set the Receive path gain (G_{42L}). The data sheet has the necessary equations to calculate these values if the system requirements for G_{42L} are defined. If assistance is needed on setting gain levels, contact your AMD field applications representative for in depth documentation on gain setting.

C_{VRX}, C_{VTX}, R_{IN}, R_{OUT1}, and R_{OUT2} are components that are for use with the QSLAC device. C_{VRX} and C_{VTX} provide AC coupling, since the SLIC devices are ground referenced inputs and outputs and the QSLAC device inputs and outputs are referenced to an internal voltage around 1.8 V. The values of C_{VRX} and C_{VTX} vary by application and are typically 0.1 µF that will generally give good results, but can be as high as 1 µF for some applications.

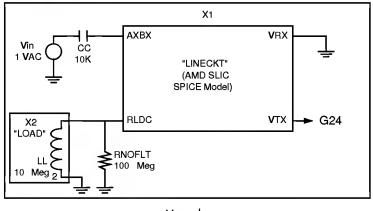
R_{OUT1} and R_{OUT2} form a voltage divider that may be needed if the signal from V_{TX} may overload the QSLAC device input. Typically, R_{OUT1} is very small and R_{OUT2} is very large.

R_{IN} is the input resistance of the QSLAC device. Consult the QSLAC data sheet for this value.

5.6 Defining The AMD SLIC G-Parameters

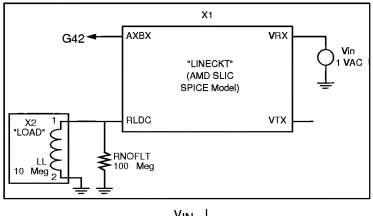
To define the G-Parameters for the SPICE model of Figure 5-5, that model must be connected (simulated) in four different arrangements. SPICE input files for each of these arrangements is built after the netlist has been created and the schematic entry program has been closed. There are two ways to get the G-parameters calculated. The first is to respond "Yes " to the prompt asking if you desire to run PSpice. If a set of G-parameters is to be calculated at some other time, the PSpice option from the SLIC menu can be chosen to calculate the G-parameters. The G-parameters will be generated using the circuits shown in Figure 5-5 through Figure 5-8. In these models, the box "LINECKT" represents the SLIC model of Figure 5-4 (FILENAME.CIR), and is treated as a subcircuit in the SPICE input file. The box designated as "LOAD" is the inductor that is connected to the R_{LDC} resistor to establish the DC loop current during the AC simulation of the circuit. The very high resistance, RNOFLOAT (R-no-float), prevents the RLDC node from floating during the SPICE simulation.

Figure 5-5 AMD SLIC G24 Model



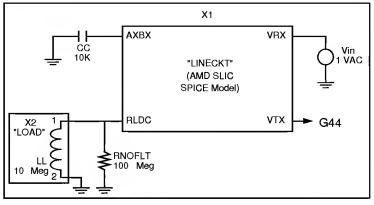
$$G_{24} = \frac{V_{TX}}{V_{IN}}\Bigg|_{V_{RX}=0}$$

Figure 5-6 AMD SLIC G42 Model



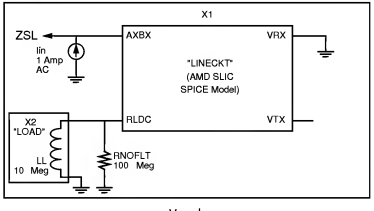
$$G_{42} = \frac{V_{IN}}{V_{RX}}\Bigg|_{I_{IN}=0}$$

Figure 5-7 AMD SLIC G44 Model



$$G_{44} = \frac{V_{TX}}{V_{RX}}\Bigg|_{V_{IN}=0}$$

Figure 5-8 AMD SLIC ZSL Model



$$Z_{SL} = \frac{V_{IN}}{I_{IN}}\Bigg|_{V_{RX}=0}$$

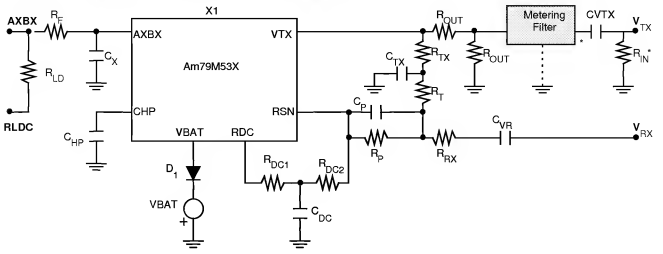
5.7 AMD Metering SLIC Models

The AMD SLIC model within the WinSLAC program provides options of choosing one of AMD's metering SLICs. However, since the actual circuit for injecting and filtering this metering information can vary depending on the specific design, no provision is made for including any of this metering interface circuitry. In order to use any of this metering circuitry, it must be added to the SLIC circuit when using the schematic entry program.

Figure 5-9 shows an example placement for inserting a metering filter into the four-wire output of the SLIC model.

To add the metering components, enter the schematic entry program as outlined previously, and enter the desired components using the schematic entry program. For help on this see the Adding Components in Schematics (Section 6.1.4) section on use of the schematic entry program. **Important:** It is necessary that nodes AXBX, RLDC, VTX, and VRX remain so named, identifying the correct input and output nodes. After editing the circuit file to include the metering filter definition, extract the netlist using the schematic entry program Analysis ➢ Create Netlist menu selection and proceed with coefficient optimization.

Figure 5-9 Metering Filter Placement for the AMD SLIC



5.8 Generic SLIC Models and Representation

The WinSLAC program does not have a built-in circuit model to represent a generic SLIC. However, a means is provided to enter or load G-parameter data that has been measured in the lab or created via simulation using the user's own SPICE file. If the G-parameter files have been created separate from the WinSLAC program, it is important that the parameters represent the same measurements. Also, if stability analysis is to be performed during coefficient calculation, each of the G-parameter files requires additional simulation. The G-parameter files consist of 3 sets of data containing 40+200+200 points. The first set is the linear sweep from 100 to 4000 Hz, the second is a decade sweep from 1 to 10000 Hz with DC loop holding, and the third is a decade sweep from 1 to 10000 Hz without the DC loop holding circuit.

5.9 Transformer SLIC Models

The WinSLAC program provides a schematic for a transformer SLIC. For more information, see the *Transformer Interface Requirements Application Note* (PID 22036). The model is made of two distinct sections: the interconnecting circuit elements external to the transformer and the transformer equivalent elements.

Figure 5-10

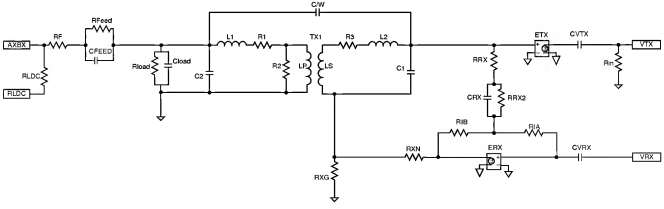
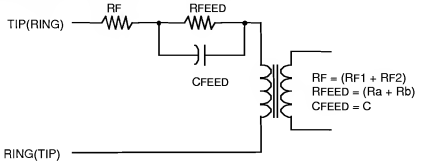


Figure 5-11 Simplified Single-Ended Line Feed Transformer Circuit



SLIC Model Interconnecting Circuit Elements:

RF	Fuse resistor DC resistance. R_F represents the total resistance of two fuse resistors and, if not used, may be set to a minimum value (i.e., $1x10^{-6}$, since SPICE must have non-zero element values).
RFeed, CFeed	Series resistance and capacitance of line circuit design, if needed to represent model.
Rload, Cload	Shunt loading across tip/ring due to any additional elements in line circuit design.
RRX, RRX2, CRX	Elements of the "Z-SLIC" which define the source impedance that is reflected through the transformer to define the two-wire impedance. (Note that this is inside the SLAC impedance synthesis feedback loop and is therefore alterable by the SLAC device.)
ETX	This represents a transmit path gain block if needed as part of the design. (The default gain is set to unity where simulation is not effected.)
ERX	This represents a receive path gain block if needed as part of the design. The gain may be positive or negative, where an inverting amplifier could be simulated. (The default gain is set to unity where simulation is not effected.)
RIA, RIB, RXN, RXG	These are resistors that are used as switches for different options of SPICE simulation. RIA and RXG default values are 1e-6 that equates to a short circuit. RIB and RXN default values are 1e+10 that equates to an open circuit. Combined with ERX gain settings, different drive configurations are available from this generic SPICE circuit topology.

Transformer Equivalent Circuit Elements

An equivalent circuit of a transformer is shown in Figure 5-10. and is provided to assist the user in visualizing the circuit elements used in the SLIC model representing the transformer circuit. The components shown represent the intrinsic values of a transformer. The correct assignment of these values is very critical for the transformer SLIC. If the values are incorrectly defined, all resulting calculations will be in error. Values may be obtained from the transformer manufacturer or through laboratory measurements.

LP	Primary (line side) winding magnetizing inductance.
LS	Secondary side winding magnetizing inductance.
KPS	Mutual inductance of LP and LS.
L1	Primary (line side) leakage inductance.
L2	Secondary (SLAC device side) leakage inductance.
R1	Primary (line side) DC (winding) resistance.
R2	Primary winding leakage resistance.
R3	Secondary (SLAC device side) DC winding resistance

C1 Secondary winding distributed capacitance.

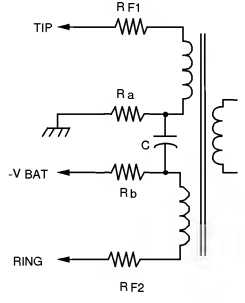
C2 Primary winding distributed capacitance.

CIW Inter-winding capacitance, primary to secondary. The default value is set to a minimum value.

Note:
The inductance values of LP and LS are used to represent the turns ratio of the transformer.

R_{FEED} and C_{FEED} are the equivalent ac elements from a DC feed network and bypass as used in a resistive line feed arrangement. Such an arrangement is shown in the following figure. (The transformer shown in these figures is representative of an actual transformer, not an ideal model). Battery feed is provided through a pair of resistors (R_a and R_b), fusing resistors are used in series with both the tip and ring leads, and a capacitor is used to bridge these signal paths across the transformer windings. A simplified, single-ended model as shown in the following figure where $C_{FEED} = C$, $R_F = R_{F1} + R_{F2}$ and $R_{FEED} = R_a + R_b$. (If the design does not require these elements, they may be set to a very small value such as $1x10^{-6}$. Again, do not set it to 0 due to SPICE's inability to use zero-valued elements.)

Figure 5-12 Balanced Line Feed Transformer Circuit



6.0 MICROSIM

6.1 Schematics Software

The WinSLAC program takes advantage of MicroSim's evaluation version of the Schematics program. This gives users a visual representation of the circuit and the flexibility to easily edit, print, and configure it to match any SLIC circuit. This section is intended to explain the use of the Schematics program to generate G-Parameter representations for the WinSLAC software. No attempt is made to completely describe all the features of the Schematics program.

Note:

The MicroSim Corporation has been purchased by OrCad, Inc.; however, the WinSLAC software only supports versions 6.0 through 8.0 of the full production version of the MicroSim software available from Orcad, Inc. (http://www.orcad.com)*. For more information or help on the features of the eval version or obtaining a full version, please contact OrCad.*

6.1.1 Creating/Modifying a SLIC Schematic

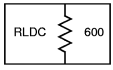
The first step in generating G-parameters is to create a schematic. This can be accomplished by selecting SLIC ➢ Create Schematic from the control bar. This will display a list of the available SLICs that can be used with the SLAC device chosen at the start of the program, see Section 7.1, SLAC Device Selection. Selecting any one of the SLICs in the pop-up list will cause a new schematic to be created with that SLIC. The WinSLAC software will then bring up a file save dialog box which requests a filename and directory to save the newly created schematic. The WinSLAC program will then invoke the Schematics program with a new schematic containing the SLIC that was selected from the pop-up list.

A previously created WinSLAC schematic can be edited from the WinSLAC program by choosing SLIC ➢ Modify Schematic. This option brings up a file open dialog box which allows you to navigate through your directories and folders to find the schematic you wish to edit. The WinSLAC software then invokes schematics with that file. Once in Schematics, whatever changes are necessary to the circuit can be made.

6.1.2 Changing Component Values

The Schematics program makes it very simple to modify the SLIC circuit. For whichever component you want to change the value of, simply double click on the number and a dialog box will appear allowing you to change its value. In the following RLDC example, you would double-click 600. The name of the resistor can also be changed in the same way. It is important to note that some names on the schematic cannot be changed or the WinSLAC software will not be able to process it. Double-clicking on the resistor itself will bring up all of its properties. The value and name can be changed within this dialog as well, but it is not as straightforward. See also Section 6.1.3, Typos When Using Schematics.

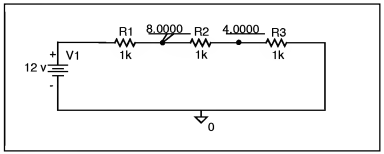
Figure 6-1 RLDC Example



6.1.3 Typos When Using Schematics

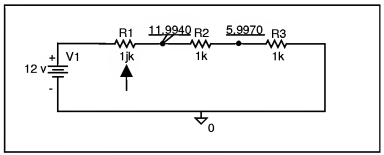
When editing the default design parameters in MicroSim's Schematics, there is an artifact of the program that will **not** detect certain types of mistakes regarding the values associated with the components. Figure 6-2 shows a normal circuit without any errors. The voltage drops are correct.

Figure 6-2 Normal Circuit



The editor will not detect certain types of typos. For example, if you happen to press the 'j' and 'k' keys together, the editor will not give an error message, but it will use the wrong value for the calculation part of the program. Figure 6-3 illustrates the point.

Figure 6-3 Circuit with Errors



Notice that the voltage drop across R1 is incorrect, due to the typo in the value field. In the WinSLAC software, the only indication you might see is if an error message is created during the compute coefficient stage that tells you the required gain of a 'G' or 'R' parameter is outside the allowable range. The WinSLAC program will set the parameter to the min or max as required and continue with the remaining computations. The charts produced will look correct, but the measured results in the lab will be completely erroneous.

Another common mistake is to use "M" with the intention of meaning 10^6. PSpice interprets "M" or "m" as 10^{-3} resulting in incorrect G-parameters. 10^6 can be represented with "MEG".

6.1.4 Adding Components in Schematics

Adding components in Schematics varies slightly depending on what version you are running. If running a version other than the evaluation version supplied with the WinSLAC software, see Section 6.1.13, Configuring Full Versions of Microsim Packages, please consult the manual for that version. In the evaluation version, a new component can be added by selecting Draw ➢ Place Part from the Schematics menu bar or by pressing the corresponding button (represented by a logic AND symbol). Doing so will bring up the parts dialog which allows you to browse the different parts available in the currently selected library. The number of parts available in the evaluation version is somewhat limited, but is sufficient for most SLIC circuits.

To select a resistor, either type R in the first dialog or select Browse. If browsing go to the analog library. There you will see a list of the passive components to choose from with a description of the one currently selected. Double click on R to place a resistor. This will drop you back to the schematic page. You can change the orientation of the component by pressing CTRL-R and CTRL-F for rotate and flip. You must then connect the component via wire, see Section 6.1.5, Drawing a Wire in Schematics.

The libraries supplied with the WinSLAC version of evaluation Schematics include:

- **analog** – Passive analog components.
- **breakout** – Parameterized devices for model purposes.
- **connect** – Connectors.
- **eval** – A collection of common diodes, transistors, etc.
- **ports** – Global ports, off-page ports, interface ports.
- **source** – Stimulus symbols.
- **slic** – Provided by AMD. Contains all publicly available SLIC models.
- **slic2** – Provided by AMD.
- **special** – Simulation command symbols.

The libraries used by the generic WinSLAC schematics are: analog, eval, ports, source, slic and slic2. The full version of MicroSim Schematics comes with a vast array of libraries with much more detailed libraries specific to particular manufacturers.

Occasionally AMD will provide updated or new SLIC libraries.

6.1.5 Drawing a Wire in Schematics

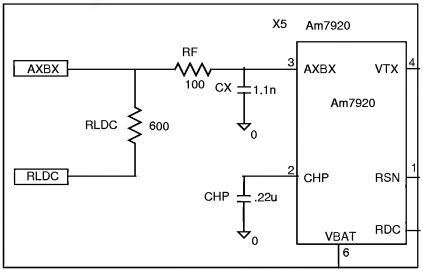
Two components can be connected by attaching a wire between them. The wire drawing mode can be entered by either selecting Draw ➢ Wire from the menu bar or by pressing the corresponding button (a thin line with a pencil). To start the wire press the left mouse button at the starting point. Then drag the mouse to the end point, or corner point and press the button again. If you end on a component, the wire will stop. If not, it will place the first section of wire and allow you to continue from there in another direction. The wire draw mode will continue until you connect to another component or discontinue the wire using the right mouse button.

6.1.6 Single Ended Model Notes (Rf & Cx)

Due to the node limitations of the evaluation PSpice, the Tip/Ring nodes of the SLIC circuits are represented as a single ended model. This means that the fuse resistor (RF) must be the sum the fuse resistance in A and B legs of the physical circuit. In addition, the CX capacitor must be the series sum of CAX and CBX (half the value of either CAX if CAX=CBX). No other deviations from the physical circuit are necessary to accommodate this model.

In Figure 6-4, the value of 100 Ω for RF represents the sum of 50 Ω from the A lead and 50 Ω from the B lead. The value of 1.1 nF for CX represents a 2.2 nF cap in the A lead and a 2.2 nF cap in the B lead.

Figure 6-4 Single Ended SLIC Model



6.1.7 Netlisting

Before leaving the Schematics program, it is necessary to create a netlist of the schematic. To do so, select Analysis ➢ Create Netlist from the menu bar. This will allow the WinSLAC software to interface to the circuit with 12 other subcircuits that are fed into PSpice. Simulation is run on the 12 circuits and then the data is organized into the four G-parameter files. It is very important that the interface node names are never changed on the schematic, otherwise the WinSLAC software will not recognize the circuit properly. See Section 6.1.10, Interface Node Names.

6.1.8 Saving a Schematic

To save a schematic within MicroSim Schematics, select File ➢ Save. This will save the schematic with the same name that it was opened as. See also Section 6.1.1, Creating/Modifying a SLIC Schematic.

To make several versions of the same schematic with just minor change, the Save As command can be used to save each of the slightly different schematics under different filenames. Be sure to create a Netlist for each of the different schematics after they have been saved as a different filename. The netlist function creates the list with the current filename, so care should be taken not to confuse which filename is currently being netlisted with which name.

6.1.9 Exiting Schematics

Schematics can be exited by selecting File ➢ Exit. If the schematic has not been saved since netlisting or changes have been made, it will ask if the file should be saved. It is recommended that you do so at that time to insure that no changes are lost.

Schematics must be exited before returning to the WinSLAC software. The WinSLAC program does not have the ability to start multiple Schematic sessions and will not operate normally if Schematics or PSpice is initiated and not exited.

6.1.10 Interface Node Names

It is very important that the four interface nodes originally on the schematic retain the same name throughout the G-parameter creation process. These are four boxed nodes on the edges of the schematic labeled AXBX, RLDC, VTX, and VRX. If any of these names is changed in any way, the WinSLAC software will no longer be able to create G-parameters with the schematic. All other component names do not matter as long as they fall within the syntax of PSpice and Schematics.

6.1.11 The SLIC Parts Libraries

Each publicly available AMD SLIC has a part model for both Schematics and PSpice. The Schematics libraries are called SLIC.SLB and SLIC2.SLB and contain 27 current AMD SLICs. All upcoming SLICs will be supplied in a supplement library that can be added to the parts choice list, see the next section, Section 6.1.12, Schematic Eval Limitations. All the SLICs are oriented the same with the exception of the ASLIC and ISLIC devices (Am79231, Am79R241, Am79R251). This means that they can easily be interchanged on the schematic sheet with only the surround component values needing to be changed.

6.1.12 Schematic Eval Limitations

The MicroSim Schematics evaluation version has a number of limitations including, but not limited to a limit to the number of parts that can be contained in a single library and the number of libraries that can be opened at the same time. Therefore, all future AMD SLICs will be provided in a second library file which may have to take the place of one of the current schematic library files which is not being used (such as the ABM.SLB). New libraries can be added in a method similar to Step 3 in the following section.

6.1.13 Configuring Full Versions of Microsim Packages

If a registered version of MicroSim Design Tools (including at least PSpice and Schematics) exists on the same computer as the WinSLAC software, the WinSLAC program can be modified to work with the full version instead of the demo version that is supplied with the WinSLAC software. In order for everything to work correctly, the following changes must be made:

1. Edit the WSLAC.INI file. This file can be found in your Windows directory. Add or edit the following lines to the end of the file:

   ```
   [MicroSim]
   PSpice=C:\msimev_8\bindl\pspicead.exe
   PSched=C:\msimev_8\bindl\psched.exe
   INI File=C:\winnt\msim_evl.ini
   ```

   Where C:\msimev_8\bindl\ is the directory containing your MicroSim executables (PSCHED.EXE, PSPICE.EXE, etc.) and C:\winnt\msim.ini is the location of the INI file for that version of MicroSim products. Different versions of PSpice may have a different executable name.

2. Enter MicroSim Schematics Editor within the WinSLAC software. This can be done by creating a schematic under the SLIC option.
   - Select Options ➢ Editor Configuration... from the menu.
   - In the Editor Configuration dialog box, there is a text box labeled "Library Path:". Edit this path statement by adding the following to the end of it:

     ```
     ;C:\winslac\lib
     ```

     Where C:\winslac is the directory into which the WinSLAC software is installed. Note that this text should be added to the path, not replace it. The semicolon IS necessary.

3. While still in the Editor Configuration dialog, choose library settings.
   - In the text box that says "Library Name:", enter the following:

     ```
     C:\winslac\lib\slic
     ```

     Where C:\winslac is the directory into which the WinSLAC software is installed.

   - Once the text is entered, select the "Add*" button. This adds the AMD SLIC library so that it is used in all schematic sessions. If the "Add Local" without the asterisk is selected, this step will need to be performed on every schematic created.
   - After pressing the "Add*" button, press OK until you are out of all dialogs.

- This step will need to be repeated for the SLIC library.

4. Select Analysis ➢ Library and Include Files... from the menu.
   - In the text box labeled "File Name:", enter the following text:

     ```
     C:\winslac\lib\slic.lib
     ```

     Where C:\winslac is the directory in which the WinSLAC software is installed.

   - Once the text is entered, select the "Add Library*" button. This adds the AMD SLIC library so that it is used in all netlist operations. If the "Add Library" without the asterisk is selected, this step will need to be performed on every schematic created.
   - Press OK to exit the dialog and return to the Schematics main screen. It is recommended that you exit the program at this time to insure that the changes made are saved to the MicroSim INI file.
5. Upon exiting, the program will ask whether you want to "..save your schematic" and "run PSPICE", choose NO for both.

The WinSLAC software should now be ready to run unhindered with the full version of PSpice. A similar procedure can be used to add any update libraries that AMD may supply in the future.

6.1.14 Metering Circuits

An advantage of the WinSLAC software over the earlier AmSLAC software is the ability to add circuitry, see Section 6.1.2, Changing Component Values. This is especially valuable for adding needed metering circuitry such as an injection circuit or a notch filter. The components that comprise these circuits can be inserted in the proper place in the SLIC circuit as shown in Figure 5-9, Metering Filter Placement for the AMD SLIC.

6.2 PSpice Software

MicroSim PSpice is a circuit analysis tool that generates the G-Parameter data for the WinSLAC software. Little interaction is required between the user and this part of the WinSLAC program. Occasionally PSpice will report an error that most likely will match one of those in the following list.

6.2.1 Errors

When PSpice encounters an error and is finished with all of the WinSLAC circuits, the WinSLAC software will return a question box asking if you would like to rename the temporary directory containing the intermediate WinSLAC circuit files. At this time you can give the directory a name of your choosing or accept the one created by the WinSLAC software. It will be necessary after exiting this dialog to either use the View function of the WinSLAC software or an editor of your choice to open up the G??.OUT file containing the error. Scan through the file for the error message. PSpice generally returns a message of why an error occurred. Following is a list of some of the more likely messages to appear with possible solutions. If the error is not in the following list, double-check any components that may have been added to the generic schematic created by the WinSLAC software.

6.2.2 Node Limit

The version of PSpice supplied with the WinSLAC software is an evaluation version, and as such has limitations on the complexity of circuits that can be analyzed. Two such limitations are the number of nodes in the circuit and the number of transistors. If such an error occurs, it will be necessary either to simplify the circuit or to use the full version of PSpice, which can be obtained by contacting MicroSim (http://www.microsim.com).

6.2.3 Disk Space

If your hard drive is nearing capacity, PSpice may not have room for the temporary files created during simulation. Possible solutions include clearing space on your hard drive by deleting unneeded files or adding a new hard drive to your system.

6.2.4 Memory

If PSpice complains that it has run out of memory, try closing any other programs that may be running. If this does not help, try rebooting and running the WinSLAC software before anything else. Another possible solution is to increase the amount of physical memory in your machine.

6.2.5 Floating Node

This very common error occurs when an inductor or capacitor is connected without a DC path. The solution to this problem is to place a very large resistor (100 MΩ) to ground at this node.

6.2.6 Bias Convergence

Occasionally a circuit will be such that without initial conditions, PSpice can not find a converging DC Bias point. This can sometimes be resolved by setting initial conditions for nodes around the nonconverging circuit.

7.0 SLAC DEVICE REFERENCE

7.1 SLAC Device Selection: DSLAC, ASLAC, QSLAC, Dual or Quad ISLAC Device

The user must choose the desired SLAC device from the menu of the different SLAC devices given on the startup of the WinSLAC software.

The AMD SLAC family controls the SLIC Devices directly, providing the user options for coefficient calculation in programming the filter and gain blocks within the device. Used in conjunction with a SLIC, the SLAC provides software configurable system solutions. The SLAC devices differ only on the level of programmability, number of channels, and added features. The common SLAC family attributes in CODEC and filtering functions include programmability of gain and frequency compensation: adaptive and transhybrid balancing and feedback from analog input to output.

- Dual ISLAC device — Am79D2251
- Quad ISLAC device — Am79Q224X
- (Advanced) ASLAC device — Am79202/03/031
- (Dual) DSLAC device — Am79C02/03/031(A)
- (Quad) QSLAC device — Am79Q02/021/031

Since SLAC device selection is given at the beginning of the program, the program must be closed and restarted in order to change the SLAC device.

Note:
For further SLAC characteristics, refer to the data sheet for your specific SLAC device.

7.1.1 Quad and Dual ISLAC Devices - Am79Q224X

The Quad and Dual ISLAC devices implement a two/four-channel universal telephone line interface. This enables the design of a single, low cost, high performance, and fully software programmable line interface card for multiple country applications worldwide. All AC, DC, and signaling parameters are fully programmable via microprocessor or GCI interfaces. Additionally, the Quad ISLAC device has integrated self-test and line-test capabilities to resolve faults to the line or line circuit. The integrated test capability is crucial for remote applications where dedicated test hardware is not cost effective.

7.1.2 (Advanced) ASLAC Device - Am79202/03/031

The ASLAC device has a single transmit and receive channel, with full software programmability for multiple country applications worldwide. When used with the ASLIC device, it becomes an ASLIC/ASLAC chip set, and integrates self-test and line-test capabilities which are crucial for remote applications. ASLAC also supports adaptive balance.

7.1.3 (Dual) DSLAC Device - Am79C02/03/031(A)

The DSLAC architecture implements two independent channels that feature adaptive transhybrid balance, and adaptive balance on (A) versions.

7.1.4 (Quad) QSLAC - Am79Q02/021/031

The QSLAC device implements four independent channels that maximize linecard density.

7.2 SLAC Filter Analysis

7.2.1 Main Function

The audio signal processing circuits of the SLAC device perform the CODEC and filter functions associated with the four-wire sections of subscriber line circuitry within a digital switching system.

Analog Gain Blocks (AX, AR), digital Gain Blocks (GX, GR), and digital filtering blocks R & X Filters/Gain Blocks provide programmable gain and frequency compensation in both the transmit and receive paths.

Programmable feedback from analog input to output provides realization of multiple two-wire termination impedances from a simple fixed termination, allowing multiple line impedance requirements to be met. Programmable transhybrid balance is accomplished with the B Filter, and optional Adaptive Balance capability is available on the DSLAC, ASLAC, Dual ISLAC and Quad ISLAC devices.

7.2.2 Analysis

Filter functions within the SLAC device are performed by a series of multiplication and accumulations, where the multiplication is accomplished by repeated multiplication and shifting and summing the result with the previous value at that summation node. The method used in the DSLAC, ASLAC and QSLAC devices is Canonic Signed Digit (CSD) multiplication, where each filter coefficient is split into a series of CSD coefficient terms. These CSD terms are entered into the SLAC device's registers where each 8-bit byte contains two CSD terms. The number of CSD terms determines the filter's actual numerical range, which is [-4 to +4] for the GX, GR, X, R, Z, and the IIR section of the B filter, and [-3 to +3] for the FIR section of the B filter. The output data from the program includes the binary (hex) byte CSD values representing the desired decimal values for the filter coefficients. The method used in the ISLAC devices is standard fractional binary format of 1.15 or 2.14.

These values from a previous program execution may be viewed using the "View" button (see Section 4.8, View-Graphs Menu), or loaded into the SLAC filter programming function using Read & Set from SLAC ➢ Globals. The coefficients so obtained may then be modified if desired for "fine tuning" of the system performance.

7.2.3 ISN

Main Function

The AISN/DISN (Analog/Digital Impedance Scaling Network) provides a programmable analog feedback gain from V_{IN} to V_{OUT} and is used to scale the value of an external SLIC Impedance. When used along with the Z filter, many different line conditions can be matched using only a single fixed impedance value. This allows the line circuit to meet the termination impedance requirements of many different specifications without any hardware changes.

When the terminating impedance is connected as part of an external line interface circuit, the gain of AISN/DISN acts to scale this external impedance, thereby modifying the actual input impedance of the SLIC.

The AISN/DISN analog/digital block has a gain, which can be varied from -0.9375 to + 0.9375 by 31 steps of 0.0625 each.

AISN/DISN - Analysis

The gain of AISN is given by:

Gain(AISN) = 0.0625[(A • 16 + B • 8 + C • 4 + D • 2 + E) - 16];
where A, B, C, D, E = 1 or 0.

An ABCDE value of "10000" is a special case where the circuitry is disabled and the SLAC device's V_{OUT} pin is internally connected to Vin with a 0 dB gain, providing digital-to-digital loopback through the device from the PCM signal paths.

An ABCDE value of "00000" specifies a gain of 0, where the AISN circuitry is disabled and out of the signal path.

The gain of DISN is given by:

$$Gain(DISN) = 0.0625[(1 - 2 \bullet DISN_4)(DISN_3 \bullet 2^3 + DISN_2 \bullet 2^2 + DISN_1 \bullet 2^1 + DISN_0)]$$

$DISN_{4-0}$ = '00000' indicates a gain of 0 (cutoff) (default).

The AISN/DISN gain can be varied from -0.9375 to +0.9375 in 31 steps of 0.0625 each. (A gain selection of zero cuts off the AISN gain block.) The desired gain within this 31-step range is chosen from the table.

Note:
For further filter characteristics, refer to the data sheet for your specific SLAC device.

7.2.4 Z Filter

Main Function

The *ZIIR* and *ZFIR* filters are programmable DSP filters that provide a digital feedback path from the analog V_{IN} to the analog V_{OUT}, and effectively work in parallel with the AISN analog gain block. Where the AISN block provides a linear (real) gain, the Z filters provide a complex gain. This provides the ability to scale the external Z_{SLIC} impedance in both real and complex domains, to meet a wide range of both real and complex impedance requirements. Adjusting the Z-filter parameters optimizes Two-Wire Return Loss (2WRL).

Z Filters - Analysis

The overall Z-filter z-transform is given by:

$$H(z) = [\ \underline{H(z)FIR} + \underline{H(z)IIR}\ ]$$

Valid values for the ZFIR and ZIIR section in the WinSLAC software are within the range of [-4 to +4] for each coefficient and entered in ascending order (Z_0 first, Z_1 second, etc.)

In the case of the Quad and Dual ISLAC devices, coefficients Z_0–Z_5 are in Q2.14 format, and ZIIR is in Q1.15 format. This gives a range of [-2 to 1.989] and [–1 to 0.999] respectively.

For the DSLAC device: the z-transform for the Z filter has a total of seven coefficients entered as signed decimal numbers. Of these, five are used for the FIR section and two are used for the IIR section.

For the DSLAC ZIIR section: the first value is the gain factor, and the second value is the denominator coefficient.

For the ASLAC/QSLAC section: the z-transform for the Z filter has a total of eight coefficients entered as signed decimal numbers. Of these, five are used for the FIR section and three are used for the IIR section.

For Quad/Dual ISLAC section: the z-transform function for the Z filter has a total of seven coefficients. Five are used for the FIR section, and two are used for the IIR section.

ZFIR Filter

ZFIR – Analysis

$$H(z)FIR = Z_0 + Z_1 \bullet z^{-1} + Z_2 \bullet z^{-2} + Z_3 \bullet z^{-3} + Z_4 \bullet z^{-4}$$

where Z_0, Z_1, Z_2, Z_3, and Z_4 are the programmable coefficient terms. The CSD's for the coefficients, Z_0–Z_4, are calculated by the program.

For DSLAC device: the hexadecimal coefficients generated in the program's output listing, correlate to the filter terms in the order of: Z_5, Z_0, Z_1, Z_2, Z_3, Z_4, and Z_6. This ordering is correct for loading the coefficients automatically into the SLACIF software when using DSLAC device and DSLAC evaluation boards.

For ASLAC/QSLAC device: the generated hexadecimal coefficients in the program output listing correlate to the filter terms in order: Z_0, Z_1, Z_2, Z_3, Z_4, Z_5, Z_6, and Z_7. This ordering is correct for loading the coefficients automatically into the SLACIF software when using ASLAC/QSLAC devices and ASLAC/QSLAC evaluation boards.

For the Quad/Dual ISLAC device: The hexadecimal coefficients generated in the program output listing correlate the filter terms in the order of Ziir, Z_5, Z_0, Z_4, Z_3, Z_2, and Z_1. This ordering is correct for loading the coefficients automatically into the SLACIF software when using the Quad/Dual ISLAC device and the Quad/Dual ISLAC evaluation boards.

ZIIR Filter

For DSLAC device: the IIR section is a 2-coefficient IIR.

For ASLAC/QSLAC device: the IIR section is a 3-coefficient IIR.

For Quad/Dual ISLAC device: the IIR section is a 2-coefficient IIR.

After initial coefficient calculation from WinSLAC program, the ZIIR coefficient values may be manually varied and subsequent recalculations of other SLAC device coefficients in order to optimize overall SLAC device/system stability if desired.

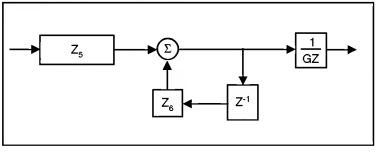
ZIIR - Analysis

For coefficient sets developed for DSLAC device, the ZIIR section of the Z filter is implemented with the following z-transform:

$$H(z)IIR = \frac{Z_5}{GZ(1 - Z_6 \bullet z^{-1})}$$; where Z_5 and Z_6 are the two programmable coefficient terms.

The DSLAC Z filter contains an additional gain block GZ and is implemented with the architecture in Figure 7-1:

Figure 7-1 DSLAC device - Z Filter Architecture



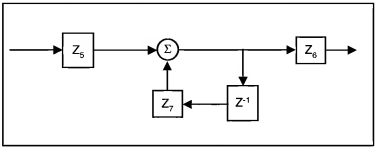
The GZ gain function allows increased filter gain to better utilize significant bits of data propagating through the IIR filter, then normalizes the gain to prohibit that added gain affecting system performance. The normalized Z_5 value is calculated by the program, then it is increased by the GZ gain factor (1, 2, 4, 8, 16, 32, 64, 128) to as high a limit as possible without danger of signal clipping. (Typical calculated values of Z_5 are around the 0.01 range, allowing for large gain increases.) The output block then attenuates the signal of the preceding IIR filter section by GZ in order to normalize the total gain through the filter.

For coefficient sets developed for ASLAC/QSLAC devices, the ZIIR section of the Z filter is implemented with the following z-transform:

$H(z)IIR = \frac{Z_5 \bullet Z_6 \bullet Z_7 \bullet z^{-1}}{1 - Z_7 \bullet z^{-1}}$; where Z_5, Z_6 and Z_7 are programmable coefficient terms.

The ZIIR section of the Z filter contains an additional gain block (Z_6) and is implemented with this architecture:

Figure 7-2 ASLAC/QSLAC Devices - Z Filter Architecture



The CSD's for the coefficients (Z_5, Z_6, and Z_7) are calculated by the program. Z_6 is used for ZIIR filter scaling only. Its value is typically: ($0 < Z_6 \geq 1$). The input to the ZIIR filter section is first increased by a gain of $1/Z_6$, improving dynamic range and avoiding truncation limitations through processing within this filter. The ZIIR filter output is then multiplied by Z_6 to normalize the overall gain. Z_5 is the actual ZIIR filter gain value defined by the programmed coefficients, but it also includes the initial ($1/Z_6$) gain. The theoretical effective ZIIR gain, without the Z_6 gain and normalization is actually Z_5/Z_6.

For coefficient sets developed for Quad/Dual ISLAC device, the ZIIR section of the Z filter is implemented with the following z-transform:

$$Hz(z)IIR = \frac{-Z_5 \bullet (Z_{iir}\, z^{-1})}{1 + Z_{iir} \bullet z^{-1}}$$

Notice the sign of the Ziir is inverted from other AMD XSLAC devices to allow Ziir = -1.

7.2.5 R Filter

Main Function

Programmable gain and frequency compensation in both the transmit and receive paths are made by the analog gain blocks (AX, AR), digital gain blocks (GX, GR), and digital filtering blocks (X, R).

This filter helps modify the frequency response for the receive path to meet the overall frequency response specifications of the system. This filter may be programmed for line equalization and to correct any attenuation distortion caused by the Z filter's action.

For the DSLAC device: the R filter is a six-tap FIR section operating at a 16 kHz sampling rate.

For the ASLAC/QSLAC/Quad ISLAC/Dual ISLAC devices: the R filter consists of a 6-tap FIR section operating at a 16 kHz sampling rate; and a single-tap IIR section operating at 8 kHz sampling rate.

R Filter Analysis

For the DSLAC device: the z-transform for the R filter has six coefficients entered as signed decimal numbers. The values are entered in ascending order (R_0, R_1, etc.)

The z-transform for the R filter is defined as:

$H_R(z) = R_0 + R_1 \bullet z^{-1} + R_2 \bullet z^{-2} + R_3 \bullet z^{-3} + R_4 \bullet z^{-4} + R_5 \bullet z^{-5}$

Valid values are within the range of [-4 to +4] for each coefficient.

The CSDs for the coefficients (R_0–R_5) are calculated by the program.

For the ASLAC/QSLAC device: the z-transform for the R filter has seven coefficients entered as signed decimal numbers. The values are entered in ascending order (R_0 first, R_1 second, etc.)

The overall R-filter z-transform is given by:

$H_R(z) = H_R(z)FIR + H_R(z)IIR$

The RFIR section of the R filter is implemented with the following z-transform:

$H_R(z)FIR = R_0 + R_1 \bullet z^{-1} + R_2 \bullet z^{-2} + R_3 \bullet z^{-3} + R_4 \bullet z^{-4} + R_5 \bullet z^{-5}$

The RIIR section of the R filter is implemented with the following z-transform:

$$H_R(z)IIR = \frac{1 - z^{-1}}{1 - R_6 \bullet z^{-1}}$$

Valid values are within the range of [-4 to +4] for each coefficient.

For the Quad ISLAC device: the z-transform for the R filter has seven coefficients entered as signed decimal numbers. The values are entered in ascending order (R_0 first, R_1 second, etc.)

The overall R-filter z-transform is given by:

$H_R(z) = H_R(z)FIR + H_R(z)IIR$

$H_R(z)FIR = R_0 + R_1 \bullet z^{-1} + R_2 \bullet z^{-2} + R_3 \bullet z^{-3} + R_4 \bullet z^{-4} + R_5 \bullet z^{-5}$

$$H_R(z)IIR = \frac{1 - z^{-1}}{1 + R_{iir} \bullet z^{-1}}$$

R_0 and R_1 are in Q2.14 format, and R_2–R_5 in Q1.15 format. RIIR is in Q1.15 format.

7.2.6 X Filter

Main Function

The X filter operates identically to the R filter; however, modifying the frequency response of the transmit path of the system.

For the DSLAC/ASLAC/QSLAC/Quad ISLAC/Dual ISLAC devices: the X filter is a six-tap FIR section and is part of the frequency response correction network of the transmit path. This filter may be programmed for line equalization and to correct any attenuation distortion caused by the Z filter's action.

X Filter Analysis

For the DSLAC/ASLAC/QSLAC/Quad ISLAC/Dual ISLAC devices: the z-transform for the X filter has six coefficients, entered as signed decimal numbers. The values are entered in ascending order (X_0 first, X_1 second, etc.)

The z-transform for the X filter is defined as:

$H_x(z) = X_0 + X_1 \bullet z^{-1} + X_2 \bullet z^{-2} + X_3 \bullet z^{-3} + X_4 \bullet z^{-4} + X_5 \bullet z^{-5}$

The coefficients (X_0–X_5) are calculated by the program.

For the DSLAC/ASLAC/QSLAC devices: Valid values are within the range of [-4 to +4] for each coefficient in CSD format.

For the Quad ISLAC/Dual ISLAC devices: X_0 and X_1 are in Q2.14 format, and X_2–X_5 are in Q1.15 format. These make up the valid values for the X filter taps.

7.2.7 Gain Blocks

Main Function

Signal gain for both transmit (analog "V_{IN}"-to-digital PCM output) and receive (digital PCM input-to-analog "V_{OUT}") signal paths within the SLAC device is independently controlled with programmable analog and digital gain blocks.

GX and AX are programmable gain blocks in the transmit path of the SLAC device; while GR and AR are programmable loss blocks in the receive path of the SLAC device.

Analog gain block AX has two gain settings of either 0 dB (unity gain) or +6.02 dB (gain of 2). The analog gain block AR also has two gain settings of either 0 dB (unity gain) or -6.02 dB (gain of 0.5). The program default for both AX and AR is 0 dB (unity gain).

The GX gain block provides a digital programmable signal gain from 0 to +12 dB, and the GR block provides a digital programmable signal attenuation from 0 to -12 dB. The GR gain can be selected in 0.1 dB increments from unity (0 dB) up to +10 dB, and above +10 dB, the step size increases to 0.3 dB for CSDs. The GR block attenuation can be selected in 0.1 dB steps throughout its 0 to -12 dB range. Gain or attenuation settings for the GX and GR blocks are entered in dB. Both GR and AR allow a programmed receive path gain range from 0 dB to -18 dB.

Gain Block Analysis

When the gain blocks are set to unity, a (0-dBm0) digital PCM signal correlates to the selected SLAC devices gain constant (volts rms) signal at the analog connection.

Note:
A/D and D/A Gain Constants of SLAC Devices are defined in the following paragraphs.

The WinSLAC software provides a menu under "Global Settings" to choose which PCM encoding and Companding method is used which in turn selects the exact gain constant. When a DSLAC device is selected, the program does not have this option and uses the same average value for both μ-law and A-law companding.

- In the DSLAC devices, a 0-dBm0 digital level corresponds to 1.550 Vrms for μ-law and 1.56 Vrms for A-law companding. The program uses the average value of 1.555 Vrms for both μ-law and A-law.
- In the ASLAC devices, a 0 dBm0 digital level corresponds to 0.586 Vrms for μ-law and 0.600 Vrms for A-law.
- In the QSLAC devices, a 0 dBm0 digital level corresponds to 0.7746 Vrms for μ-law and 0.7804 Vrms for A-law.
- In the ISLAC devices, a 0 dBm0 digital level corresponds to 0.5027 Vrms for μ-law and 0.4987 Vrms for A-law.

In DSLAC, ASLAC, and QSLAC devices the GX and GR gain setting coefficients are determined by four CSDs, each occupies two 8-bit bytes. In ISLAC devices, the GX and GR gain setting coefficients are based on Q1.15 format.

Note:
It is possible to calculate coefficients to program up to +14 dB of gain for GX, but performance characteristics above the data sheet specified +12 dB limit are not guaranteed and operation in this region is not recommended.

7.2.8 B Filter

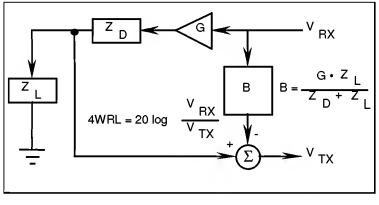
Main Function

The B (balance) filter is a programmable DSP filter used to cancel the received signal that passes across the hybrid into the transmit path, to provide transhybrid balancing in the loop. The B filter transfer function should match as closely as possible the transfer function of the echo path.

The SLAC device B filter is designed to work with preprogrammed coefficients or with coefficients determined by an adaptive algorithm. The adaptive mode uses some of the preprogrammed coefficients and generates some new ones with an internal echo-minimizing algorithm. Adaptation only applies to the FIR section of the filter.

The B Filter provides a digital feedback path from the digital receive to the digital transmit (PCM) paths. Digital PCM signals presented to the SLAC device are passed to the analog two-wire system interface through the externally connected SLIC. The far-end receiving equipment receives this now-analog signal. Analog signals transmitted from the far-end equipment share the same two-wire line interface and are passed back through the SLIC and SLAC device to its transmit PCM bus. In order to distinguish this far-end transmitted signal from the near-end signal (analog output of the SLAC device), this near-end signal must be subtracted from that appearing at the two-wire interface. This "echo cancellation" process provides the transhybrid balance for the SLAC devices line circuit design, but is dependent on proper impedance matching at the two-wire interface. The B filter can introduce a wide range of both real and complex gain, providing flexibility to cancel near-end signals over multiple system and impedance conditions. A simplified example of the four-wire echo cancellation provided by the B filter is shown in Figure 7-3.

Figure 7-3 Echo Cancellation with the B Filter



B Filter Analysis

The z-transform for the B filter has nine coefficients, entered as signed decimal numbers. Of these, eight are used for the FIR section, and one is used for the IIR section. The FIR values are entered in ascending order (B_0 first, B_1 second, etc.)The overall B-filter z-transform is given by:

$H_B(z) = [\ H_B(z)FIR + H_B(z)IIR\ ]$

Valid values are: Within the range of [-3 to +3] for the FIR coefficients, and [-4 to +4] for the IIR denominator coefficient.

An IIR denominator coefficient greater than 0.9 may cause oscillation. A negative denominator value is prohibited.

For the DSLAC device: The z-transform for the BFIR filter is defined as:

$H_B(z)FIR = B_0 + B_1 \bullet z^{-1} + B_2 \bullet z^{-2} + B_3 \bullet z^{-3} + B_4 \bullet z^{-4} + B_5 \bullet z^{-5} + B_6 \bullet z^{-6}$

The z-transform for the BIIR filter is defined as:

$$H_B(z)IIR = \frac{B_7 \bullet z^{-7}}{1 - B_8 \bullet z^{-8}}$$

The CSD's for the coefficients (B_0 through B_8) are calculated by the program, but are either not used or overwritten during Adaptive Balance operation. For the ASLAC and DSLAC devices, not all coefficient values that can be generated by the program are compatible with the adaptive mode. If the choice to use the adaptive mode is made, the B-filter coefficients should be manually set to "seed" values for adaptive operation. An example of these values can be found in the SLAC device data sheet.

For the ASLAC/QSLAC device: the FIR values are entered in ascending order from (B_2 first, B_3 second, etc.)

The z-transform for the BFIR filter is defined as:

$$H_B(z)FIR = B_2 \bullet Z^{-2} + B_3 \bullet Z^{-3} + \ldots + B_8 \bullet z^{-8}$$

The z-transform for the BIIR filter is defined as:

$$H_B(z)IIR = \frac{B_{10} \bullet z^{-10}}{1 - B_{11} \bullet z^{-1}}$$

For the ASLAC/QSLAC device: the first two values of the time response, (B_0, B_1), are ignored in the WinSLAC software, to present the two delay taps in ASLAC device that DSLAC device does not have. The CSD's for the coefficients (B_2–B_{11}) are calculated by the program, but are either not used or overwritten during Adaptive Balance operation. For the ASLAC device, not all coefficient values generated by the program are compatible with the adaptive mode. If the choice to use the adaptive mode is made, the B-filter coefficients should be manually set to "seed" values for adaptive operation. An example of these values can be found in the SLAC device data sheet.

For the Quad ISLAC device, the z-transform for the B filter is defined as:

$$H_B(z) = B_2 \bullet z^{-2} + B_3 \bullet z^{-3} + B_4 \bullet z^{-4} + B_5 \bullet z^{-5} + B_6 \bullet z^{-6} + B_7 \bullet z^{-7} + B_8 \bullet z^{-8} + B_9 \bullet z^{-9} + \frac{B_{10} \bullet z^{-10}}{1 + B_{IIR} \bullet z^{-1}}$$

For the Dual ISLAC device, the z-transform for the B Filter is defined as:

$$H_B(z) = B_2 \bullet z^{-2} + B_3 \bullet z^{-3} + \ldots + B_{13} \bullet z^{-13} + \frac{B_{14} z^{-14}}{1 + B_{IIR} \bullet z^{-1}}$$

All B filter taps are in the Q1.15 format. The sign of the Biir is inverted from other AMD XSLAC devices to allow Biir = -1.

A.0 WinSLAC PROGRAM FILE OVERVIEW

The WinSLAC software includes a number of individual files, which are added at installation, plus other files that are created as a result of the program execution. These files are summarized in this section for reference. The files that are installed to hard disk by the installation program includes the executable programs, and data files used by the WinSLAC software, plus supplementary files which are also included with the software distribution. Details are provided in this section for files, which are produced by running the WinSLAC software.

Temporary Files

Temporary files are created during program operation, containing data for every user entry field of the program, as well as the path and file names.

Some temporary files are created in the directory containing the schematic file during the G-parameter calculations. These include the Schematic ".CIR", ".NET", and ".ALS" netlist files. These three files are created when the netlist is created within Schematics. All three of these files are deleted after computation of G-parameters to avoid confusion of modifications made to the schematic. The WinSLAC software also creates a temporary directory in the directory containing the schematic file. This directory contains the ".CIR" circuit files, ".OUT" output files, and ".DAT" temporary G-parameter files. These are normally deleted upon successful conclusion of the G-parameter PSpice calculations. If an error occurs during Pspice, this temporary directory will remain allowing the user to view the ".OUT" files to locate the error.

*The *.PAR Parameter File*

[file_name].PAR: The parameter files (.PAR) contain all user-entry data that is entered (or already resides within) into the System Parameter menus and the SLAC menus of the program. It is written only through the *File ➢ Save Parameter File* or *File ➢ Save As Parameter File* as menu selection. The WinSLAC program can read the parameter files, so that information that is initially entered and saved can be again used for subsequent program executions.

*The *.ARF Output file*

[output_name].ARF: This is one of the key output files created by the WinSLAC program. It contains the calculated data for each of the programmable filters of the SLAC device and describes the expected performance of the system design.

The .ARF file provides the user information on the state of the filter settings, which filters are enabled, disabled, or set for calculation, and gives the name of the SLIC G parameter file; as well as, the name and state of the System parameter (*.PAR) files. Any problems that may occur during coefficient computation and calculation, will be stated in this file along with a description on the course of action taken, or it will state that it was not possible to correct the situation.

The ARF output file can be read into the WinSLAC program from various locations. Since it contains the values for coefficients, which were calculated from its initial creation, those values can then be used to preset coefficients into any or all of the SLAC device's coefficient registers.

*The *.HEX Output File*

[output_name].HEX: This is a brief summary file of only the hexadecimal data for each of the SLAC device registers. Its data duplicates that within the .ARF file, and this smaller file is provided for convenience to define the data, which would be written into the device by the system controller.

Note:
The format for AISN data is not actually represented in hex format. This is also true of that portion of the ARF output file.

*The *.PRF Predicted Results File*

[output_name].PRF: This is a "Predicted Results File" which lists in tabular numeric format the data that is displayed graphically within the .ARF file in order to describe the system's performance.

*The *.G24, .G42, .G44, .ZSL G-Parameter Files*

[slic_name].G24, .G42, .G44, .ZSL: These are the four G-parameter files which are created from the PSpice simulation of the SLIC model. The base name [slic_name] is the same as was defined by the user for the SLIC subcircuit file at the time SPICE was selected within the SLIC menu. For further detail on G-parameters and their generation, refer to Section 4.5.4, G-Parameter Entry.

*The *.BOD Stability (Bode) File*

[output_name].BOD: This is a BODE data file with tabular data of magnitude and phase vs. frequency describing each of the three closed loop conditions for which stability checking and compensation is performed. This file is helpful in locating areas of instability created by the impedance synthesis loop of signal paths through the SLIC and signal paths through the AISN and Z-filter of the SLAC device. The three conditions checked for stability is for the two-wire port of the linecard design: 1) Terminated with the design's two-wire impedance; 2) Open circuited; and 3) Short-circuited.

*The *.DRF Desired Response File*

[output_name].DRF: The "Desired Response File" lists the tabular numeric data describing each filter's independent "desired" response in terms of magnitude and phase as a function of frequency. This is the data input in the frequency domain, which the computation algorithms in the WinSLAC program try to match in the digital domain for each individual filter. This data may be helpful in cases where external dsp filter computation may be implemented if desired.

*The *.ZIN Impedance File*

[output_name].ZIN: This is a tabular data file of magnitude and phase vs. frequency representing the design's synthesized two-wire actual impedance. It is this data which is compared to each frequency's value of the "Zd" setting which describes the program's predicted two-wire return loss. The contents of this file provide a predicted value of the actual impedance appearing across the tip/ring terminal.

B.0 EXAMPLES

The examples in this section are designed to take you through the process of creating and evaluating a linecard design using the WinSLAC program. Each example examines the process using a specific SLAC device.

B.1 AM7920 and QSLAC Devices

The Am7920 and QSLAC device were selected as target devices due to their popularity and widespread use. While it is possible to use a different order, the sequence followed here is the most logical in terms of a new design. There are seven main steps to creating and evaluating a design:

1. Specify the system impedance parameters using the *System* menu options. Section 4.2, System Menu, has an explanation of what these parameters are and what types of options are available. As an option, you can load a .PAR file, which contains information such as: desired line, termination and load impedances; 2 and 4 wire return loss; transmit and receive path attenuation distortion templates; equalization templates; and predetermined filter settings. These settings determine the target market for the design.
2. Run the schematic capture program to create a schematic file and the associated netlist files.
3. Use the PSpice program to perform SPICE simulations of the circuit and create the G-parameter files that represent the 7920 SLIC device.
4. Specify the desired filter settings.
5. Compute filter coefficients and predicted response charts.
6. Examine the predicted response charts.
7. Adjust filter values as needed and recompute.
8. Exit the WinSLAC program when finished.

Starting the WinSLAC program from program manager

* Choose a SLAC device from the option box. For this example, select QSLAC in the *Select SLAC* dialog box. To load the .PAR file, choose *File ➢ Load System Parameters*. For this example, use the qslac20.par file. This .PAR file targets the North American market with 900 ohms in series with 2.16 uF.

Selecting a SLIC device and root file name

* Pull down the *SLIC* menu.

* Select the 7920 from the SLIC list box.

Selecting a root and schematic file name

* Type in the path and file name to save your schematic under. The path name is optional. The name you choose will be used as a root file name in various other sections of the program. This step creates a schematic file with a .SCH extension. If you want to use a separate directory or folder, create it before starting the WinSLAC software. The editor will not let you create a new one when choosing a root file name.

Editing the schematic capture page for the SLIC circuit

* Change the desired component values using the techniques described in Section 5.5, Component Selection.

* When finished, select *Analysis ➢ Create Netlist*.

* If you check the directory and folder containing the root schematic name, you will see three additional files. They will have the root name plus these extensions: .CIR, .ALS and .NET. These

are the files PSpice uses to analyze the circuit and are used in the next phase of the program. A complete list of file extensions and their meanings is located in Appendix A.

* Choose *File* ➢ *Exit* and either save or abandon edits.

The Run PSpice dialog box

* If you want to run PSpice, click Yes. The PSpice simulation creates about 36 temporary files that are deleted automatically upon successful completion. The final files created at this step use the root file name with the following extensions: .ZSL, .G24, .G42 and .G44. These files are used in the next steps to create the desired response file and filter coefficients. Once PSpice has completed an error free run, the three files mentioned in step three above are deleted from that directory. These files are deleted to prevent multiple versions of the design. If you edit the text netlist and do not update the schematic, then you lose correlation between the schematic and the results. That is one reason to save your edits of the schematic; it makes it easier to recreate the netlist if you want to run the program again.

* If you do not want to run the PSpice simulation when exiting the schematic editor, select No. If you do not run PSpice upon exiting the schematic capture, the program will save the three files created in the same directory as the schematic you created previously. These files will remain until you do run PSpice. To run PSpice later, select *SLIC* ➢ *Run PSpice* in the main WinSLAC screen.

* The netlist uses the same name as the schematic file. If you forgot to create a netlist, there will not be any .CIR files for you to select from in the *Select Circuit File to Open* dialog box.

Options under the SLAC menu

* There are four filter menus under the *SLAC* menu. They are: AISN-Z filter, R&X filters, B filter and Adaptive Balance and Global Settings.

* Each filter has three options available. They can be selected to Set, Calculate or Disable. Choose:

- Set to define a response pattern.
- Calculate to have the WinSLAC software calculate the desired response.
- Disable to remove the filter from the circuit.

The last option is helpful when evaluating SLIC responses separately.

* The Global options may be of interest. You can force the WinSLAC software to compute all the desired responses and coefficients.

Using the Compute menu to compute the coefficients for the circuit

* The program will create six permanent files upon successful completion of the Compute command. They are: .BOD, .ZIN, .ARF, .DRF, .HEX and .PRF. You may also see a warning message upon completion, the warning messages are included in the .ARF file. Chose *File* ➢ *View* to browse the .ARF file.

When the program finishes computing the coefficients, the graphs of the predicted performance and templates will be displayed. Stability correction tends to indicate a poor ZT match.

Making changes in the SLIC circuit

* Select *SLIC* ➢ *Modify Schematic.*

* Double click on the schematic path/name you created previously.

You can use the 7920 data sheet or reference design as an aid in calculating ZT, ZRX, etc.

ZT=250

*(Z2wireIN-2

*Rf) Z2wireIN =900 Ohms Rf=50 Ohms ZT=200k Ohms

ZL=Z2wireIN =900 Ohms G42L=1 ZRX=ZL/G42L

* (500

*ZT)/[ZT+250(ZL+2Rf)] =128.6k Ohms

* Make sure all other components match the test setup in Note 1 of the Data Sheet or the values that you have calculated to meet your requirements.

* Save any changes you make in the schematic.

* It might be a good idea to change the root name of the schematic when you make these changes. If you do not, the new calculations based on this schematic will overwrite the previous results. This will not allow you to use the Compare feature of the WinSLAC software to easily determine if your changes made the response better or worse. The Compare feature is described below.

Creating a new netlist, run PSpice and recompute the filter coefficients

* If the GR is outside the 0 to -12dB range, it tends to indicate the receive gain of the SLIC is incorrect. The GR gain varies inversely with the receive gain.

* Adjust the gain of the SLIC, try using G42L=2.

* Repeat the computation steps again.

If the response is still unacceptable, experiment with resistor network values to continue modifying the response of the circuit.

- Multiple passes can be examined together using the Compare feature of the chart display screen. While the current results are displayed, click the *Compare On* button which opens the *Specify File to Compare With* dialog box. Select the .PRF file of interest and it will be plotted on the same chart as the current results. By turning compare on and off, you can look at many different results without rerunning all the simulations. Compare will only allow you to see two results at a time. It is also available under the *View-Graphs* menu in the WinSLAC main screen. You can pick any two .PRF files you wish and compare them side-by-side.

B.2 AM79R79 and DSLAC Devices

The Am79R79 and DSLAC devices were selected as target devices due to their popularity and widespread use. While it is possible to use a different order, the sequence followed here is the most logical in terms of a new design. There are seven main steps to creating and evaluating a design:

1. Specify the system impedance parameters using the *System* menu options. Section 4.2, System Menu in the WinSLAC software itself has the explanation of what these parameters are and what types of options are available. To load an existing G-parameter file (.PAR), Select *File ➢ Load System Parameters* and, using the directory tree, choose the desired .PAR file. For this example, you can use the dslac79.par file. This sets the four main templates and system impedances to the default values.
2. Run the schematic capture program to create a schematic file and the associated netlist files.
3. Use the PSpice program to perform SPICE simulations of the circuit and create the G-parameter files that represent the 79R79 SLIC device.
4. Specify the desired filter settings.
5. Compute filter coefficients and predicted response charts.

6. Examine the predicted response charts.
7. Adjust filter values as needed and recompute.
8. Exit the WinSLAC software when finished.

Starting the WinSLAC software from program manager

* Choose a SLAC device from the option box. For this example, select DSLAC device in the *Select SLAC* dialog box.

Loading a parameter (.PAR) file

* Select *File* ➢ *Load System Parameters.*

* Using the directory tree, select the desired .PAR file. For this example, you can use dslac79.par. As mentioned above, the .PAR file sets the system parameters including ZD, ZL, and ZT. It also contains the templates for 2 and 4 wire return loss, attenuation distortion templates for transmit and receive paths, transmit and receive equalization (if used), stability check on/off, and any predefined filter settings. For this example, all the default values are used, so loading the .PAR file is optional. The other included example design file is based on the QSLAC device and is targeted for the North American market and uses a different .PAR file.

Selecting a SLIC device and root file name

* Pull down the SLIC menu.

* Select the 79R79 from the SLIC list box that appears.

Selecting a root and schematic file name

* Type in the path and file name to save your schematic under. The path name is optional. The name you choose will be used as a root file name in various other sections of the program. This step creates a schematic file with a .SCH extension. If you want to use a separate directory or folder, create it before starting the WinSLAC software. The editor will not let you create a new one when choosing a root file name.

Editing the schematic capture page for the SLIC circuit

* Change the desired component values using the techniques described in Section 5.5, Component Selection.

* When finished, Select *Analysis* ➢ *Create Netlist.*

* If you check the directory/folder containing the root schematic name, you will see three additional files. They will have the root name plus these extensions: .CIR, .ALS and .NET. These are the files PSpice uses to analyze the circuit and are used in the next phase of the program. A complete list of file extensions and their meanings is located in Appendix A.

* Choose *File* ➢ *Exit* and either save or abandon the edits.

The Run PSpice dialog box

* If you want to run PSpice, click *Yes.* The PSpice simulation creates about 36 temporary files that are deleted automatically upon successful completion. The final files created at this step use the root file name with the following extensions: .ZSL, .G24, .G42 and .G44. These files are used in the next steps to create the desired response file and filter coefficients. Once PSpice has completed an error free run, the three files mentioned in step three above are deleted from that directory. These files are deleted to prevent multiple versions of the design. If you edit the text netlist and do not update the schematic, then you lose correlation between the schematic and the results. That is one reason to save your edits of the schematic; it makes it easier to recreate the netlist if you want to run the program again.

* If you do not want to run the PSpice simulation when exiting the schematic editor, select *No*. If you do not run PSpice upon exiting the schematic capture, the program will save the three files created in the same directory as the schematic you created previously. These files will remain until you do run PSpice. To run PSpice later, select *SLIC ➢ Run PSpice* in the main WinSLAC screen.

* The netlist uses the same name as the schematic file. If you forgot to create a netlist, there will not be any .CIR files for you to select from in the *Select Circuit File to Open* dialog box.

Options under the SLAC menu

* There are four filter menus under the *SLAC* menu. They are: AISN-Z filter, R&X filters, B filter and Adaptive Balance and Global Settings.

* Each filter has three options available. They can be selected to Set, Calculate or Disable. Choose:

- Set to define a response pattern.
- Calculate to have the WinSLAC software calculate the desired response.
- Disable to remove the filter from the circuit.

The last option is helpful when evaluating SLIC responses separately.

* The Global options may be of interest. You can force the WinSLAC software to compute all the desired responses and coefficients if you need to.

Using the Compute menu to compute the coefficients for the circuit

* The program will create six permanent files upon successful completion of the Compute command. They are: .BOD, .ZIN, .ARF, .DRF, .HEX and .PRF. You may also see a warning message upon completion, the warning messages are included in the .ARF file. Chose *File ➢ View* to browse the .ARF file.

When the program finishes computing the coefficients, the graphs of the predicted performance and templates will be displayed. Stability correction tends to indicate a poor ZT match.

Making changes in the SLIC circuit

* Select *SLIC ➢ Modify Schematic*.

* Double click on the schematic path/name you created previously.

You can use the 79R79 data sheet or the AMD reference design (Part # AM79DSLAC/RSLIC) as an aid in calculating ZT, ZRX, etc.

* Make sure all other components match the test setup of the data sheet or the values that you have calculated to meet your requirements.

* Save any changes you make in the schematic.

* It might be a good idea to change the root name of the schematic when you make these changes. If you do not, the new calculations based on this schematic will overwrite the previous results. This will not allow you to use the Compare feature of the WinSLAC software to easily determine if your changes made the response better or worse. The Compare feature is described below.

Creating a new netlist, run PSpice and recompute the filter coefficients

* If the value of AISN seems large or near the maximum value, RX or TS gains are near the allowable limits or the predicted response is not as smooth as you would like, check the reference design mentioned above and/or verify the component calculations done previously for accuracy.

* Repeat the computation steps again.

If the response is still unacceptable, experiment with resistor network values to continue modifying the response of the circuit.

* Multiple passes can be examined together using the Compare feature of the chart display screen. While the current results are displayed, click the *Compare On* button which will open the *Specify File to Compare With* dialog box. Select the .PRF file of interest and it will be plotted on the same chart as the current results. By turning compare on and off, you can look at many different results without rerunning all the simulations. Compare will only allow you to see two results at a time. It is also available under the *View-Graphs* menu in the WinSLAC main screen. You can pick any two .PRF files you wish and compare them side by side.

B.3 AM79r241 and Quad ISLAC Devices

While it is possible to use a different order, the sequence followed here is the most logical in terms of a new design. There are seven main steps to creating and evaluating a design:

1. Specify the system impedance parameters using the *System* menu options. Section 4.2, System Menu, has an explanation of what these parameters are and what types of options are available. As an option, you can load a .PAR file, which contains information such as: desired line, termination and load impedances, 2 and 4 wire return loss, Transmit and receive path attenuation distortion templates, equalization templates, and predetermined filter settings. These settings determine the target market for the design.
2. Run the schematic capture program to create a schematic file and the associated netlist files.
3. Use the PSpice program to perform SPICE simulations of the circuit and create the G-parameter files that represent the 79r241 SLIC device.
4. Specify the desired filter settings.
5. Compute filter coefficients and predicted response charts.
6. Examine the predicted response charts.
7. Adjust filter values as needed and recompute.
8. Exit the WinSLAC software when finished.

Starting the WinSLAC software from program manager

* Choose a SLAC device from the option box. For this example, select Quad ISLAC device in the *Select SLAC* dialog box. To load the .PAR file, choose *File ➢ Load System Parameters*. For this example, use the Islic900.par file. This .PAR file targets the North American market with 900 ohms.

Selecting a SLIC device and root file name

* Pull down *SLIC* menu.

* Select the 79r241 from the SLIC list box that appears.

Selecting a root and schematic file name

* Type in the path and file name to save your schematic under. The path name is optional. The name you choose will be used as a root file name in various other sections of the program. This step creates a schematic file with a .SCH extension. If you want to use a separate directory or folder, create it before starting the WinSLAC software. The editor will not let you create a new one when choosing a root file name.

Editing the schematic capture page for the SLIC circuit

* Change the desired component values using the techniques described in Section 5.5, Component Selection.

* When finished, select *Analysis ➢ Create Netlist*.

* If you check the directory/folder containing the root schematic name, you will see three additional files. They will have the root name plus these extensions: .CIR, .ALS and .NET. These are the files PSpice uses to analyze the circuit and are used in the next phase of the program. A complete list of file extensions and their meanings is located in the help section of the WinSLAC software.

* Choose *File* ➢ *Exit* and either save or abandon edits.

The Run PSpice dialog box

* If you want to run PSpice, click *Yes*. The PSpice simulation creates about 36 temporary files that are deleted automatically upon successful completion. The final files created at this step use the root file name with the following extensions: .ZSL, .G24, .G42 and .G44. These files are used in the next steps to create the desired response file and filter coefficients. Once PSpice has completed an error free run, the three files mentioned in step three above are deleted from that directory. These files are deleted to prevent multiple versions of the design. If you edit the text netlist and do not update the schematic, then you lose correlation between the schematic and the results. That is one reason to save your edits of the schematic. It makes it easier to recreate the netlist if you want to run the program again.

* If you do not want to run the PSpice simulation when exiting the schematic editor, select *No*. If you do not run PSpice upon exiting the schematic capture, the program will save the three files created in the same directory as the schematic you created previously. These files will remain until you do run PSpice. To run PSpice later, select *SLIC* ➢ *Run PSpice* in the main WinSLAC screen.

* The netlist uses the same name as the schematic file. If you forgot to create a netlist, there will not be any .CIR files for you to select from in the *Select Circuit File to Open* dialog box.

Options under the SLAC menu

* There are four filter menus under the *SLAC* menu. They are: AISN-Z filter, R&X filters, B filter and Adaptive Balance and Global Settings.

* Each filter has three options available. They can be selected to Set, Calculate or Disable. Choose:

- Set to define a response pattern.
- Calculate to have the WinSLAC software calculate the desired response.
- Disable to remove the filter from the circuit.

The last option is helpful when evaluating SLIC responses separately.

* The Global options may be of interest. You can force the WinSLAC software to compute all the desired responses and coefficients if you need to.

Using the Compute menu to compute the coefficients for the circuit

* The program will create six permanent files upon successful completion of the Compute command. They are: .BOD, .ZIN, .ARF, .DRF, .HEX and .PRF. You may also see a warning message upon completion, the warning messages are included in the .ARF file. Chose *File* ➢ *View* to browse the .ARF file.

When the program finishes computing the coefficients, the graphs of the predicted performance and templates will be displayed. Stability correction tends to indicate a poor ZT match.

G.0 GLOSSARY

The following is a list of terms and their, Definitions as used in this manual. These terms describe parameters as they apply to SLIC and SLAC device functions.

A-law — A digital compression and expansion standard used in the European telephone network.

AISN — The Analog Impedance Scaling Network is a programmable analog gain of -0.9375 to +0.9375 from V_{IN} to V_{OUT}. Scaling AISN together with the Z filter allows for matching of many different line conditions with a single linecard.

ASLAC — Advanced Subscriber Line Interface Circuit

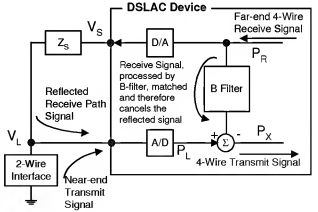
Anti-sat — Anti-saturation region. The region in a SLIC's DC feed curve that allows for signal overhead into very long loops.

Attenuation Distortion — Attenuation distortion specifies the amount of gain or loss at a specific frequency within the voice (200 Hz to 3400 Hz) band, relative to the absolute signal level at 1004 Hz.

AX and AR — Programmable analog gain (AX) and attenuation (AR) blocks in the transmit and receive analog signal paths respectively. The analog gain (loss) is selectable at unity gain, or a gain (loss) of two (one-half).

B Filter — A FIR and single pole IIR programmable filter in the SLAC device connecting from the digital (PCM) input (receive) path to the digital output (transmit) path. It is used to cancel the portion of the receive signal that travels across the hybrid into transmit path in order to minimize echo. The adaptive mode that allows the filter to actively minimize the echo is controlled by the EPG and ELT register values. For optimum four-wire to two-wire conversion with maximum echo cancellation, the signal out of the B filter, should completely cancel the portion of the receive signal, P_R in this simplified diagram, which appears at P_L out of the A/D converter.

Simplified Echo Cancellation by the B Filter



Battery — A term for the main power supply for two wire telephone circuits. Classically battery powered with a potential between -42 Vdc and -57 Vdc.

BORSCHT — Acronym. "Battery feed, Overvoltage tolerant, Ringing, Supervision, Code/decode, Hybrid and Test. Referring to functions required by a linecard.

CO	Central Office. The switching equipment that provides local exchange telephone service for a given geological area.
CODEC	Code/Decoder. Component in a linecard that performs the A/D and D/A conversion of the transmission paths. Includes some form of compression and expansion function also.
Compander	Slang for COMpressor/exPANDER that refers to the method of compression and expansion used in PCM coding.
CSD	Canonic Signed Digit (CSD) multiplication is used within the SLAC device to represent the programmable block coefficient values that are used to define the numeric values of that block's function. In this scheme, a multiplication is accomplished by repeatedly shifting the multiplicand and summing the result with the previous value at that summation node.
Current Feed	The sourcing of power from a SLIC that is characterized by a constant programmed current. This will have an anti-sat region.
Desired Impedance(ZD)	The desired impedance is the nominal input impedance of the exchange, and is specified by the PTT. This is the impedance looking into and presented by the TIP and RING terminals. This impedance value is established by the physical ZSLIC impedance connected externally to the SLAC device, *and* its transformation due to the feedback paths through the AISN and Z filter blocks. This parameter can be entered into the WinSLAC program as a ratio of polynomials, or by amplitude and phase values entered from the frequency range of 100 Hz to 4000 Hz in steps of 100 Hz.
DISN	The Digital Impedance Scaling Network is a programmable analog gain of -0.9375 to +0.9375 from V_{IN} to V_{OUT}. Scaling DISN together with the Z filter allows for matching of many different line conditions with a single linecard.
DSLAC	Dual Subscriber Line Audio-Processing Circuits.
Echo Path Gain(EPG)	A programmable value that predicts the amount of the receive signal traveling across the hybrid into the transmit path. The EPG is used to cause the B filter to stop adapting when the pre-determined level of (transmit) near end (double talker) signal is detected. The adapting process stops with the B filter using its last frozen values when: $\frac{PL}{\text{(maximum of last 8 samples of PR)}} > EPG$
Error Level Threshold (ELT)	A programmable value that establishes the trans-hybrid loss that the B filter (ELT) will attempt to meet when its adaptive balance mode is enabled. The B filter stops adapting when the residual error signal is below the value set by the ELT. This error signal represents the amount of the digital receive signal (P_R in the figure) that appears after the summation node (P_X) that the B filter has not completely removed from the V_L signal across the analog two-wire load. The adapting process stops with the B filter using its last frozen values when: {bmc BMP00062.BMP}< ELT.

Four-Wire Return Loss	Four-Wire Return Loss (4WRL), also referred to as "Transhybrid Loss", is a (4WRL) measurement of the load matching to the line circuit's internal "hybrid" (two-wire to four-wire) balance network. This hybrid network is designed to match externally applied line (two-wire) impedance. With optimum matching, a signal presented to the four-wire input of this balance network will be passed completely to the two-wire load and no reflections will occur to be passed back to the four-wire output. Any mismatch however will result in a portion of the four-wire input being reflected to the four-wire output.
G-Parameters	A SLIC can be represented by a network consisting of three gain blocks and an impedance block, where these four total blocks are referred to as G-parameters. The gain and impedance characteristics of the SLIC can be fully defined in terms of its measurable voltage or current values at the four-wire input, four-wire output, and two-wire port connections. The G-parameters, identified as: G24, G42, G44, and ZSL, are each specified in tabular form as gain and phase values at established frequencies.
GX and GR	Programmable digital gain (GX) and attenuation (GR) blocks in the transmit and receive digital signal paths respectively.
Hybrid Function	The function of two to four wire conversions performed on a linecard. This will include two-wire impedance synthesis and longitudinal balance.
ISLAC	Intelligent Subscriber Line Audio-Processing Circuits.
ISN	The Impedance Scaling Network is the combination of AISN/DISN with the Z-filter. The programming of these blocks results in different linecard impedances allowing for matching of many different line conditions with a single linecard.
Line (or Balance)	The line or balance impedance is the impedance that is presented to the Impedance (ZL) TIP/RING terminals of the line. This is the impedance to be matched by the linecard to obtain optimum trans-hybrid balance and is specified by the PTT.
Linecard	The circuit that interfaces the two wire subscriber to a digital backplane. Includes all of the functions in BORSCHT.
Longitudinal	Synonym for common mode.
Longitudinal Balance	The transmission parameter that ensures two wire common mode signals are not converted into transmission signals.
Loop	The DC circuit formed by a SLIC sourcing a length of two wires terminated by some DC impedance (telephone typically 400).
μ-law	A digital compression and expansion standard used in the North American and Japanese telephone networks.
Metallic	Synonym for differential.
Metering	Toll Metering. A method of monitoring phone call tolls for subscribers. Metering is provided by the linecard in the form of 12 or 16 kHz tone burst (in various international countries).
PABX or PBX	Private (Automatic) Branch Exchange. Provides switching for private application with trunks to public infrastructure.
PCM	Pulse Code Modulation. Referring to the method of digital transmission from this type of CODEC.

Polarity Reversal	A method of signaling on a two wire loop. The DC potential is reversed with (POL REV) respect to tip and ring.
POTS	Plain Old Telephone Service. Basic telephone service provided to general public.
PSTN	Public Switched Telephone Network. The complex network of telephone exchanges that allows for calls from state to state and country to country.
PTO	Public Telecommunications Operator.
QSLAC	Quad Subscriber Line Audio-Processing Circuits.
R Filter and X Filter	Programmable 6-tap FIR filters within the SLAC device, one each for the receive (R filter) and transmit (X filter) path, used to adjust and flatten the system frequency response.
Receive Path	The receive path signal originates as a digital PCM signal presented to the SLAC device's digital input, where after digital processing and digital-to-analog conversion, it is passed as an analog signal through the V_{RX} output to the SLIC. After being transferred to the two-wire interface through the SLIC, it appears as a "received" signal at a station set connected to the two-wire line. The receive or transmit path designation is established at the reference point of an apparatus connected to the system's two-wire interface.
Receive Relative Level	The relative receive level is the analog level of a 1 kHz signal that is produced at (LO) the two-wire interface at its termination impedance when a 0 dBm0 digital signal is presented at the PCM interface. This value is referenced as "L_o" in the CCITT transmission recommendations.
Resistive Feed	The sourcing of power from a SLIC that is characterized by a programmed source resistance. Will have an anti-sat region and current limit region. AKA: Voltage feed.
Return Loss Templates	The minimum 2/4-WRL (in dB) that is acceptable for the line interface is (for 2WRL and 4WRL) established by the PTT. Graphical "templates" describe the limits with the return loss limits defined over a function of frequency. In the WinSLAC program, the dB limits are entered in 100 Hz steps, from 100 to 4000 Hz.
SLIC	The term SLIC as used in the SLAC device or WinSLAC software is defined as "Subscriber Line Interface Circuit," which identifies the entire circuit connecting the four-wire analog port of the SLAC device to any externally connected analog interface. The SLIC may be comprised of any active or passive circuitry, as well as a solid state (transformer-less) line interface, which may itself be referred to as a "SLIC" by various manufactures.
Subscriber	A telephone customer.
System Parameters	System Parameters identify the performance and operational characteristics of the telephone line two-wire interface. These parameters include impedance specifications, return loss requirements, and receive and transmit frequency response and signal levels.

Termination Impedance

The termination impedance is the test source (generator) and load (signal (ZT) measurement) impedance used when performing two-wire line measurements. This is usually the same as the desired impedance but may be different as specified by the applicable design requirements or PTT. (For example, Bellcore's LSSGR specifies a two-wire input impedance of 900 • + 2.16 µF, and frequency response to be measured with a 900 • source impedance.)

Tip and Ring

The two connections at the linecard that source the two wire interface.

Transhybrid Balance

The parameter of a line circuit that describes its ability to cancel the reflection from the two wire interface. In other words, 'four wire return loss' (4WRL).

Transmit Path

The transmit path signal originates from the station set's transmit (microphone, etc.) signal source, travels across the 2 wire interface to the SLIC, is passed to the SLAC device's VTX analog input, and then to the device's digital PCM output.

Transmit Relative Level

The relative transmit level is the analog level of a 1 kHz signal that is needed to (Li) produce a 0 dBm0 digital signal at the PCM output. This value is referenced as "Li" in the CCITT transmission recommendations.

Two-Wire Interface

A length of two wires that carries duplex transmission and DC signaling between a SLIC and some terminal device (i.e. telephone).

Two-Wire Return Loss

Two-Wire Return Loss (2WRL) is a measurement of the impedance match (2WRL) between the two-wire TIP/RING interface and the characteristic impedance of the connected line. It is measured in dB and is defined as:

$$\text{Return Loss}_{dB} = 20 \log_{10} \left| \frac{Z_D + Z_L}{Z_D - Z_L} \right|$$

If the impedance's were exactly equal the return loss would be infinite. Any mismatch between a source and the interface will result in an echo reflection being returned back to the source.

Z Filter

A six-tap FIR and single pole IIR programmable filter, providing a digital feedback path from the analog (transmit) input to the analog (receive) output. This feedback path, along with that of the AISN block, provides programmable feedback to the externally connected SLIC to transform an external physical (Z_{SLIC}) impedance into a different (Z_D) nominal impedance.

Z_{SLIC}

The Z_{SLIC} is the physical impedance that appears at the SLAC device's analog output and serves as the connection to the SLIC. This impedance, transformed by the feedback action of the AISN and Z filter blocks, is used to establish the realized nominal impedance of the system, Z_D.

I.0 INDEX

APPENDIX E
COPY OF ALTERA CORPORATION,
"FIR COMPILER MEGACORE FUNCTION" ("SOLUTION BRIEF 41")

FIR Compiler MegaCore Function

Solution Brief 41 **June 1999, ver. 1**

Target Applications:
Cellular base stations, spread-spectrum communications, set-top boxes, and several other digital signal processing (DSP) applications

Family:
APEX™ 20K, FLEX® 10K, FLEX 8000, and FLEX 6000

Ordering Code:
PLSM-FIR

Vendor:

101 Innovation Drive
San Jose, CA 95134
http://www.altera.com
Tel. (408) 544-7000

Features

- Fully integrated finite impulse response (FIR) filter development environment
- Highly optimized for Altera® device architectures
- Supports parallel or serial arithmetic architectures
- Supports any number of taps
- Includes a built-in coefficient generator
- Imports integer or floating-point coefficients from third-party tools
- Supports multiple coefficient scaling algorithms
- Supports coefficient widths from 4 to 32 bits of precision
- Supports signed or unsigned input data, with widths from 4 to 32 bits wide
- Permits user-selectable output precision via rounding and saturation
- Determines symmetry and selects appropriate architecture automatically
- Creates MATLAB Simulink, VHDL, and Verilog HDL simulation models
- Generates Quartus™ and MAX+PLUS® II vector files
- Includes an impulse, step function, and random input testbed
- Provides dynamic resource estimates

General Description

Many digital systems use signal filtering to remove unwanted noise, to provide spectral shaping for communications channels, or to perform signal detection or analysis. FIR filters are used in systems that require a linear phase and have an inherently stable structure. Typical filter applications include signal preconditioning, band selection, and low-pass filtering.

The filter design process involves identifying coefficients that match the frequency response specified for the system. The coefficients determine the structure of the filter. You can change which signal frequencies pass through the filter by changing the coefficient values or adding more coefficients.

A fully parallel, pipelined FIR filter implemented in a programmable logic device (PLD) can operate at data rates above 100 megasamples per second (MSPS), making PLDs ideal for high-speed filtering applications. The FIR compiler MegaCore™ function has an interactive wizard-driven interface that allows you to create custom FIR filters easily. The wizard outputs simulation files for use with third-party tools, including MATLAB Simulink. The FIR compiler MegaCore function speeds up the design cycle by:

- Finding the coefficients needed to design custom FIR filters.
- Generating clock-cycle-accurate FIR filter models (also known as bit-true models) in the Verilog HDL and VHDL languages, and for the MATLAB environment (M-files and Model Files).
- Automatically generating the code required for the Quartus or MAX+PLUS II software to synthesize high-speed, area-efficient FIR filters of various architectures.
- Creating standard test vectors (i.e., impulse, step, and random input) to test the FIR filter's response.

MegaCore

Functional Description

You can run the FIR compiler wizard using the Quartus or MAX+PLUS II MegaWizard™ Plug-In Manager. After you set the function's parameters, the wizard generates a customized function that can be instantiated in your design file. Table 1 describes the options for the FIR compiler wizard.

Table 1. FIR Compiler Wizard Options

Page	Description
Input Data Type	The width of the input data bus (from 4 to 32 bits wide). Specifies a signed or unsigned bus.
Coefficients	The filter coefficients can be read from a file or generated using the FIR compiler wizard. In both cases, you can scale the coefficients and indicate the precision bits. The wizard detects the filter symmetry and selects an appropriate architecture. The function lets you specify the sample rate (either in Hertz or in relation to the Nyquist rate), the number of taps, and cut-off frequencies. The function supports low-pass, high-pass, band-pass, and band-reject filters. The function also supports rectangular, Hanning, Hamming, and Blackman windows. As you change the coefficient settings, you can view the frequency and response of the filter dynamically.
Limiting Precision	The function determines the output bit width based on the actual coefficient values and the input bit width. You can reduce your filter's precision by removing bits from the most significant bit (MSB) via truncation or saturation, or from the least significant bit (LSB) via truncation or rounding.
Architecture	You can indicate whether the filter is parallel or serial, and the number of channels for the filter. APEX 20K and FLEX 10K devices contain embedded system blocks (ESBs) and embedded array blocks (EABs), respectively, which are ideal for use with serial filters. Both parallel and serial filters allow pipelining, which lets you tradeoff between area and speed.
Simulation Output Files	The function generates several types of simulation files, including MAX+PLUS II Vector Files (.vec), MATLAB and Simulink models, MATLAB testbench files, Verilog HDL models, and VHDL output files.

Figure 1 shows the coefficient generator and MegaWizard Plug-In for the FIR compiler MegaCore function. From the coefficient generator, you can specify the function's sample rate, number of taps, cut-off frequency, filter type, and window method.

Figure 1. FIR Compiler Coefficient Generator

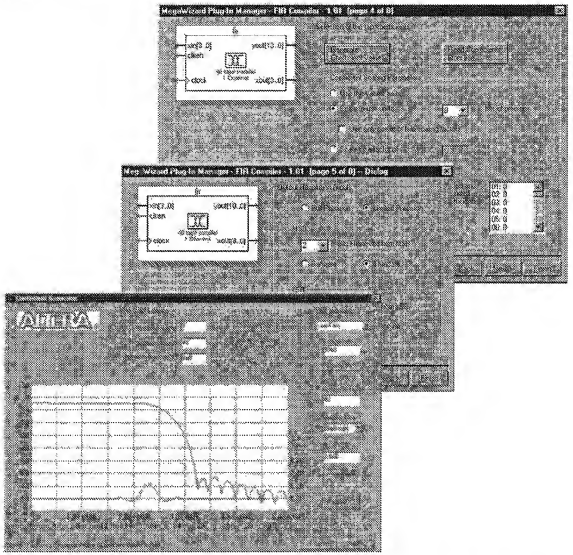
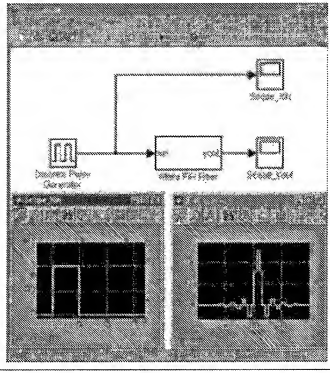


Figure 2 shows the MATLAB Simulink interface for the FIR compilet function.

Figure 2. System-Level Simulation with the MATLAB Simulink Interface



Performance

Table 2 describes the logic element (LE) requirements for the FIR compiler MegaCore function.

Table 2. FIR Compiler Performance

Device	Speed Grade	Parameters	Utilization		Performance (MHz)
			LEs	EABs	
FLEX 10KE	-1	17-tap fully parallel filter	879	0	82
		19-tap fully parallel filter	1,260	0	101
		79-tap serial filter	761	5	69

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com

APPENDIX F
COPY OF U.S. PATENT APPLICATION
PUBLICATION NO. 2005/0047779 ("JAYNES")

US 20050047779A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2005/0047779 A1**

Jaynes et al. (43) **Pub. Date: Mar. 3, 2005**

(54) **POST-DETECTION, FIBER OPTIC DISPERSION COMPENSATION USING ADJUSTABLE INVERSE DISTORTION OPERATOR EMPLOYING TRAINED OR DECISION-BASED PARAMETER ADAPTATION (ESTIMATION)**

(75) Inventors: **Lonnie Scott Jaynes**, Melbourne, FL (US); **Jerry Busby Wood**, Melbourne, FL (US)

Correspondence Address:
Allen, Dyer, Doppelt, Milbrath & Gilchrist, P.A.
1401 Citrus Center
255 South Orange Avenue
Post Office Box 3791
Orlando, FL 32802-3791 (US)

(73) Assignee: **HARRIS CORPORATION**, MELBOURNE, FL (US)

(21) Appl. No.: **10/653,316**

(22) Filed: **Sep. 2, 2003**

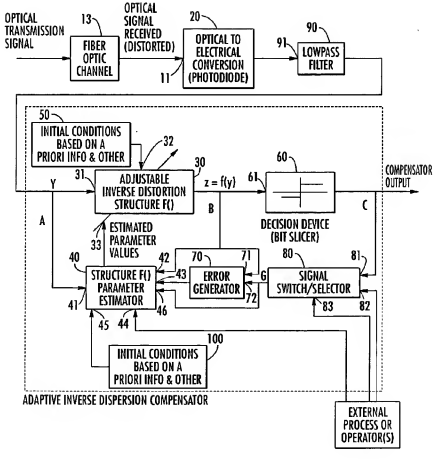
Publication Classification

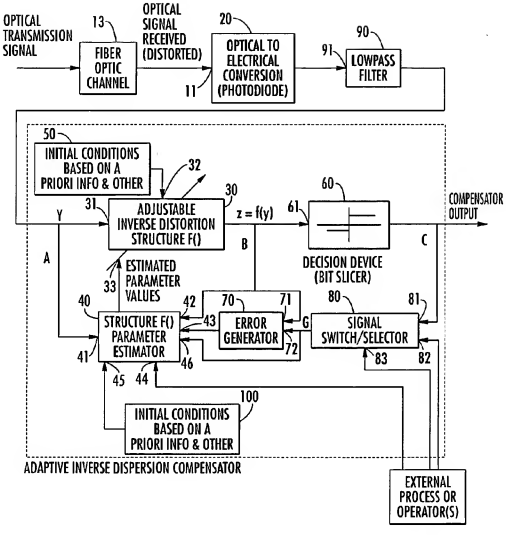
(51) **Int. Cl.**[7] **H04B 10/08**

(52) **U.S. Cl.** **398/29**

(57) **ABSTRACT**

An adaptive system composed of an adjustable inverse distortion operator or structure compensates for dispersive distortion in a fiber optic channel. Parameters of the inverse distortion operator are automatically estimated and updated in accordance with minimizing some cost function of an error signal obtained by differentially combining the output of the inverse distortion operator with downstream decision values or with an undistorted training signal. Undistorted training signals may be derived from bit patterns (e.g., preamble) expressly transmitted for the purpose of adjusting the compensation system or from other non-training patterns known to be embedded in the received signal from the transmitter.




OPTICAL TRANSMISSION SIGNAL
13
FIBER OPTIC CHANNEL
OPTICAL SIGNAL RECEIVED (DISTORTED)
20
OPTICAL TO ELECTRICAL CONVERSION (PHOTODIODE)
11
90
91
LOWPASS FILTER
50
INITIAL CONDITIONS BASED ON A PRIORI INFO & OTHER
32
30
31
ADJUSTABLE INVERSE DISTORTION STRUCTURE F()
z = f(y)
60
61
COMPENSATOR OUTPUT
Y
A
B
C
DECISION DEVICE (BIT SLICER)
ESTIMATED PARAMETER VALUES
33
40
42
70
71
80
81
STRUCTURE F() PARAMETER ESTIMATOR
43
ERROR GENERATOR
G
SIGNAL SWITCH/SELECTOR
41
46
72
83
82
45
44
100
INITIAL CONDITIONS BASED ON A PRIORI INFO & OTHER
ADAPTIVE INVERSE DISPERSION COMPENSATOR
EXTERNAL PROCESS OR OPERATOR(S)

POST-DETECTION, FIBER OPTIC DISPERSION COMPENSATION USING ADJUSTABLE INVERSE DISTORTION OPERATOR EMPLOYING TRAINED OR DECISION-BASED PARAMETER ADAPTATION (ESTIMATION)

FIELD OF THE INVENTION

[0001] The present invention relates in general to communication systems and subsystems, and is particularly directed to a method and an apparatus for compensating for dispersive distortion in a communication channel, particularly a fiber optic channel, by means of an adaptive system composed of an adjustable inverse distortion operator or structure that is installed in an electrical signal processing path of an opto-electronic receiver, wherein the parameters of the inverse distortion operator are automatically estimated and updated in accordance with an error signal obtained by differentially combining the output of the inverse distortion operator with downstream decision values or with an undistorted training signal.

BACKGROUND OF THE INVENTION

[0002] A number of communication networks and systems, such as, but not limited to high data rate fiber optic communication systems, employ communication channels that are dispersive—in that they cause the energy of a respective signal component to be dispersed or spread in time as it is transported over the channel. In an effort to reduce the effects of dispersion, some fiber optic systems predistort the signal in a manner that is intended to be "complementary" to the effect of the optical channel, so that "optimally" at the receiver the original signal, prior to the predistortion operation, may be recovered. Other systems address the problem by dealing directly with the channel itself, such as by using dispersion compensating fibers (DCFs). These approaches can be difficult or expensive to apply under various conditions and, from a functional and architectural standpoint, are relatively rigid, in that they tend to be easily affected by operational changes or by environmental changes, such as mechanical vibration or variations in temperature. In addition, the desire to use channel multiplexing (e.g., wavelength division multiplexing (WDM)) in fiber optic cables, increased data rates, and longer uninterrupted cable lengths complicate and exacerbate the deficiencies of traditional compensation schemes.

SUMMARY OF THE INVENTION

[0003] In accordance with the present invention, problems of conventional methodologies for dealing with channel dispersion in a. high data rate fiber optic communication system, such as those described above, are effectively obviated by means of a post-detection adaptive system composed of an adjustable inverse distortion operator or structure that is inserted in the electrical signal processing path of the output of an opto-electronic converter or detector, wherein the parameters of the inverse distortion operator are automatically estimated and updated in accordance with minimizing some cost function of an error signal obtained by differentially combining the output of the inverse distortion operator with downstream decision values for a "decision-based" parameter estimation or adaptation mode, or with an undistorted training signal for a "trained" parameter estimation or adaptation mode.

[0004] In the decision-based adaptation mode, parameter values are estimated based upon an error signal formed between the output of the inverse distortion structure and the output of a bit slicer (binary decision device) that is coupled to the output of the structure. An error generator performs a prescribed differential combining operation on its two inputs and supplies an error signal to a parameter estimation unit that is representative of the difference, dissimilarity, or variance between structure-processed data and decisions produced by the bit slicer. In this mode, parameter estimates for the inverse distortion structure (operator) can be updated on a continuous basis as data is received and processed by enabling a parameter estimation process realized or contained inside the parameter estimation unit. The onset time and duration of estimation processing is determined by the value of an estimator control signal (enable/disable/type) generated by a process or operator external to the compensator and coupled to the estimation unit.

[0005] In the trained adaptation mode, parameter values are estimated based upon an error signal formed between the output of the inverse distortion operator and a corresponding but undistorted training signal, both of which are supplied to the error generator. In the trained adaptation mode, training signals are derived either from patterns expressly transmitted for the purpose by the upstream transmitter or from other known or predictable patterns not explicitly transmitted for compensator adjustment or adaptation. As in the decision-based adaptation mode, the estimator control signal is used to enable/disable (or gate) the parameter estimation (update) process. In trained adaptation mode, activation and deactivation of the parameter update process is synchronized with the detection and availability of training signal data. Training signal data (undistorted) is generated by an external process or operator and is time-aligned with corresponding data received and processed by the compensator.

[0006] In addition to using one of the modes described above exclusively, the invention may apply both parameter estimation modes together (and possibly others), with each mode being activated over different time intervals under the control of an external process or operator. In this combined mode of operation, structure parameter estimates can be updated according to range of different criteria and schedules depending on the type and quality of data available so as to optimize overall compensator performance.

DESCRIPTION OF THE DRAWINGS

[0007] The single Figure diagrammatically illustrates a preferred, but non-limiting embodiment of the adaptive inverse distortion compensator and parameter estimation mechanisms of the present invention.

DETAILED DESCRIPTION

[0008] Before describing in detail the adaptive inverse distortion compensation method and system of the present invention, it should be observed that the invention resides primarily in prescribed modular arrangements of conventional digital communication circuits and associated digital signal processing components and attendant supervisory control circuitry therefor, that controls the operations of such circuits and components. In a practical implementation that facilitates their being packaged in a hardware-efficient equipment configuration, these modular arrangements may

be readily implemented in different combinations of field programmable gate arrays (FPGAs), application specific integrated circuit (ASIC) chip sets, microwave/millimeter-wave monolithic integrated circuits (MIMICs), and digital signal processing (DSP) cores.

[0009] Consequently, the configuration of such arrangements of circuits and components and the manner in which they are interfaced with other communication equipment have been illustrated in the drawings by a readily understandable block diagram, which shows only those specific details that are pertinent to the present invention, so as not to obscure the disclosure with details which will be readily apparent to those skilled in the art having the benefit of the description herein. Thus, the block diagram illustration is primarily intended to show the major components of the invention in a convenient functional grouping, whereby the present invention may be more readily understood.

[0010] Attention is now directed to the single Figure, wherein a preferred, but non-limiting, embodiment of the present invention is diagrammatically illustrated as comprising an input port **11**, to which an optical communication signal, such as that transported over a dispersive optical fiber **13**, is coupled. As a non-limiting example, the optical communication signal may comprise a conventional synchronous optical network (SONET)-based signal, such as the SONET STS-192 signal, which contains 384 frame synchronization bytes (a priori known) in each 125-microsecond time interval (192 A1 octets and 192 A2 octets).

[0011] Input port **11**, consisting of a suitable optical coupler (not shown), is coupled to an opto-electronic receiver unit, such as a photodiode detector **20**, which converts the received optical communication signal into an electrical signal. This electrical signal is representative of the optical communication signal as received from the dispersive optical fiber and, as such, contains both the desired but unknown information signal component as well as auxiliary known information, such as framing components of the optical communication signal, as well as any (dispersive) distortion that has been introduced into the optical communication signal as a result of its transport over the fiber optic channel **13**.

[0012] The output of the photodiode **20** is coupled to the input port **91** of a lowpass filter **90**. The lowpass filter is used to suppress undesirable out-of-band (high-frequency) components present in the photodiode output signal. The output of the lowpass filter **90** is coupled to an input port **31** of a controllably adjustable inverse distortion structure or operator **30** and to an input port **41** of a parameter estimation unit **40**. The inverse distortion structure is a parameterized operator designed to implement an approximate inverse function of one or more prescribed distortion mechanisms to which the optical signal is subjected as it is propagated over the fiber optic channel. Example fiber optic distortions of interest include chromatic dispersion (CD) and polarization mode dispersion (PMD). CD has a frequency response that can be represented by the transfer function H(f)=exp{−j*k*f**2}, where f is the cyclic frequency (cycles/second), j is the imaginary unit, and k is a coefficient or parameter of dispersion. As a result, an inverse distortion structure for CD could be implemented as an operator designed to approximate the transfer characteristic Hinv(f)=exp{j*k*f**2}, where unknown values of parameter k are to be estimated and adjusted by the parameter estimation process of the compensator. First order PMD can be modeled as a two-component, signal multipath process. For example, first order PMD can be represented in the time domain by the expression y(t)=gamma*x(t−t0)+(1-gamma)*x(t−t0−t1), where y(t) is the received (electrical) signal, x(t) is the signal transmitted over the PMD channel, gamma is a signal (amplitude) splitting ratio, and t1 the time delay difference between received multipath signal (polarization) components. Parameters gamma and t1 are normally variable over location and time. In this case, a suitable inverse distortion structure could be implemented as a filter (possibly requiring stabilization) whose transfer characteristic approximates the inverse of the one implied for PMD above with parameters gamma and t1. Similar to the inverse CD case previously described, unknown values for parameters gamma and t1 would be estimated and supplied by the parameter estimation process of the compensator. It should be noted that actual parameters to be estimated by the parameter estimation unit might ultimately depend upon, to some extent, the specific structure selected to approximate the inverse distortion process. For example, the inverse distortion structure may require estimates for functions of distortion parameters instead of the parameters themselves. Under ideal conditions, the inverse distortion operator **30** might be expected to completely remove targeted distortion effects imparted on the transmitted signal as a result of its propagating the FO channel. In reality though, since parameter estimates and the inverse distortion structure of the compensator are approximations, complete distortion elimination would not normally be expected. However, the invention is designed to significantly reduce signal distortion by taking advantage of a priori knowledge about the form of the distortion and by including a mechanism to automatically find and track changing or unknown distortion parameter values.

[0013] The adjustable inverse distortion operator **30** is coupled to an associated memory **50**, which holds and supplies to the operator initial values based upon a priori knowledge of channel distortion characteristics as well as other initializing data. The inverse distortion operator **30** has its output coupled to a (binary) decision device or bit slicer **60**, the output of which delivers the detected data stream with distortion compensation. The output of the inverse distortion operator is additionally coupled to the structure parameter estimator unit **40** and to the first input port **71** of an error generator unit **70**. Error generator **70** has a second input port **72** coupled to the output of a signal switch/selector unit **80**. Signal switch **80** connects one of its two input ports, **81** or **82**, with its output port according to predefined signal values appearing on its switch control input port **83**.

[0014] The training signal is a prescribed pattern that is known to the compensator, and may comprise a training preamble that is transmitted from the upstream transmitter at predefined intervals. A copy of this training signal is stored in the compensator (or accompanying receiving system) and can be used during or after a time the signal is transmitted by the transmitter to adjust or adapt parameter estimates of the compensator's inverse distortion structure according to current or prevailing channel conditions. The signal used in this regard by the compensator need not be a training signal as such, however. It may correspond to some other a priori known or predictable bit pattern that is transmitted by the transmitter.

[0015] As a non-limiting example, such a priori known bit patterns may correspond to the consecutive frame synchronization patterns or octets that occur in SONET data, referenced above. In order to take advantage of such data for compensator training purposes, frame synchronization octets would first be detected in the received data stream (or some derivative thereof) and then time-aligned or synchronized with undistorted versions of the synchronization octet patterns. Synchronized distorted and undistorted versions of signals based on the detected synchronization patterns could then be processed by the parameter estimation unit **40** and the error generator **70** (through the signal selector **80**) at process-determined times to update inverse distortion parameter values. This process of detecting and synchronizing known signal patterns for the purpose of automatically adjusting or adapting compensator/equalizer coefficients (parameters) is of the type described in our co-pending U.S. patent application Ser. No. 10/462,559, filed on Jun. 16, 2003, entitled: "Updating Adaptive Equalizer Coefficients Using Known or Predictable Bit Patterns Distributed Among Unknown Data" (hereinafter referred to as the '559 application, assigned to the assignee of the present application and the disclosure of which is incorporated herein.

[0016] Relative to the present invention, training signals (undistorted) and associated control ("gating") signals are generated by an external process or operator (perhaps similar to the one described in application xxx) and coupled to input port **82** of the signal switch **80** and input port **44** of the parameter estimation unit **40**, respectively. Received signals containing corresponding channel-distorted patterns useful for parameter estimation are coupled to input port **41** of the parameter estimation unit and to input port **31** of the inverse distortion structure **30**. Control signals are synchronized with the occurrence of detected training patterns and are used to gate the operation of the parameter estimation unit. Under the control of a "selection" signal coupled to input port **83** of the signal switch, undistorted training signals are connected to input port **72** of the error generator and input port **46** of the parameter estimator unit. It is important to note that different levels of buffering or delay (not shown in figure) may be required along signal paths A, B, and C in order to achieve proper compensator operation. These required buffers or delays could be incorporated in selected compensator components such as the parameter estimation unit, error generator, and signal switch.

[0017] Error generator **70** differentially combines signals present on its input ports **71** and **72** and places the resulting error signal (a difference, dissimilarity, or variance signal) on its output port that is coupled to the error input port **43** of the parameter estimator unit **40**. Parameter estimator unit **40** is coupled to an associated memory **100**, which holds and supplies to the unit initial values based upon a priori knowledge of channel distortion characteristics as well as other initializing data. The parameter estimator unit implements or contains an algorithm or operator designed to find parameter values for the inverse distortion structure that minimize some cost function (or expectation of some cost function) of the error signal. Example cost functions include squared error, absolute error, and uniform error. Signals coupled to input ports **42** and **46** of the parameter estimator unit could be used as an alternative to the direct error signal coupled to input port **43**. Also, depending on the specific algorithm or operator employed and in addition to the error signal, the parameter estimator may require as input the input signal of the inverse distortion structure **30**. This signal is coupled to input port **41** of the parameter estimation unit. The estimator unit generates updated parameter values for the adjustable inverse distortion structure whenever the unit is activated by an estimator control signal coupled to input port **44**. Control of the parameter estimation process may be done in accordance with different criteria including the successful detection and availability of suitable compensator training patterns. With sufficient internal buffering included on input data paths, the parameter estimator unit can be designed or configured to update parameter values at rates lower than the filtering rate of the inverse distortion structure itself. This partial decoupling of structure filtering and structure parameter estimation improves the compensator's flexibility and eases overall implementation considerations. Additionally, the parameter estimator unit may contain special functions or operators for "whitening" or synthesizing new data from raw input signal data that is better suited for estimation processing.

[0018] In the decision-based parameter adaptation (or estimation) mode, structure parameter values are estimated and updated in accordance with output decisions from decision device **60**. This mode is entered by selecting decision device **60** output via signal switch **80** and enabling the parameter estimation unit **40** using a control signal coupled to its input port **44**. In the trained parameter adaptation (or estimation) mode, structure parameter values are estimated and updated in accordance to known signal patterns. This mode is entered by using signal switch **80** to select a known "training" signal (undistorted) coupled to input port **82** and by enabling the parameter estimation unit as before. In this mode of operation, the estimator control signal is used to enable/disable the operation of the parameter estimation unit **40** according to different criteria including the occurrence (or availability) and duration of training signal data. As described above, training signals may be composed of bit patterns transmitted expressly for the purpose of adjusting compensator parameters to existing channel conditions or they may be composed of other bit patterns known to occur in the received data stream, such as the frame synchronization octets of SONET.

[0019] The inverse distortion operator parameter update mechanism of the present invention operates as follows for its respective decision-based and trained adaptation modes.

[0020] Decision-based Parameter Adaptation (Estimation) Mode

[0021] As pointed out briefly above, in this mode of operation parameter estimates are based upon a comparison of the output of the inverse distortion structure **30** with the output of the binary decision device **60**. As an electrical signal is output from the photodiode detector **20** it is coupled to the lowpass filter **90**. The output of the lowpass filter is coupled to the adjustable inverse distortion structure **30**. The output of the inverse distortion structure is coupled to the binary decision device **60** and to the first input port **71** of the error generator **70**. The output of the decision device is coupled to the error generator through the signal switch/selector **80**. Note that in the decision-based adaptation mode, the signal switch connects the output of the binary decision device to the second input port **72** of the error generator. In the trained adaptation mode, the signal switch connects an externally generated training signal (undistorted) to input

port **72** of the error generator. The error generator differentially combines the signals coupled to its input ports and supplies an error signal to the parameter estimation unit **40**. The parameter estimation unit implements or embodies an algorithm or operator designed to minimize some cost function of the error signal. Depending on the specific estimation process employed, the parameter estimation unit may also require as input, the input signal of the inverse distortion structure. In this estimation mode, parameter values can be updated more or less continuously by simply enabling the parameter estimator unit with the estimator control signal coupled to input port **44**. The estimator control signal is generated by an external process or operator and may be used to disable the estimation process under different conditions including the reception of poor or unusable data.

[0022] Trained Parameter Adaptation (Estimation) Mode

[0023] As pointed out above, in this mode of operation parameter estimates are based upon a comparison of the output of the inverse distortion structure **30**, which is coupled to the first input port **71** of error generator **70**, with an undistorted training signal coupled to the second input port **72** of the error generator and supplied through the signal switch **80** from its input port **82**. In this mode of operation, an external process or operator identifies the occurrence of known bit patterns in the received data stream and generates training signals based on undistorted versions of these patterns along with a corresponding synchronized "gating" signal (estimator control signal) and supplies these signals to input port **82** of the signal switch and input port **44** of the parameter estimator unit **40**, respectively. The estimator control signal is used to enable/disable or "gate" the operation of the estimator unit in-accordance with different criteria including the occurrence and/or availability of training data. Known bit patterns embedded in the received data stream may correspond to patterns transmitted for the express purpose of adjusting or adapting the compensator or they may correspond to other patterns known to occur in the data stream, such as the frame synchronization octets of SONET.

[0024] Combined and Other Adaptation Modes

[0025] It should be noted that other parameter adaptation modes can be defined for the compensator in addition to the baseline types described above. For example, a "blind parameter adaptation" mode can be established for the compensator by including or implementing a process in the parameter estimator unit that does not make use of independent reference signals such as training signals. Instead, parameter values are estimated based upon measured or derived quantities (for example, statistics) of selected signals such as the input signal of the compensator. Blind parameter adaptation or estimation would be activated in a manner similar to that. of other adaptation modes using the estimator control signal to enable the parameter estimator unit and to select the appropriate estimation process.

[0026] In addition to using one of the adaptation modes described above on an exclusive basis, the present invention can be applied to take advantage of combining two or more parameter adaptation modes operating together, with each being activated at different times under the control or supervision of an external process or operator. In this combined mode of operation, an external process or operator controls or supervises the mix and durations of decision-based, trained and other adaptation through the use of the compensator's signal switch/select and estimator control signals. Combined mode (or multimode) operation provides a mechanism for optimizing compensator performance over a wide range of operating conditions.

[0027] While we have shown and described several embodiments in accordance with the present invention, it is to be understood that the same is not limited thereto but is susceptible to numerous changes and modifications as known to a person skilled in the art, and we therefore do not wish to be limited to the details shown and described herein, but intend to cover all such changes and modifications as are obvious to one of ordinary skill in the art.

What is claimed is:

1. For use with a system for processing a communication signal that has been transported over a dispersive communication channel, so as to recover an unknown information signal contained in said communication signal, wherein said communication signal is represented as an electrical communication signal, a method of processing said electrical communication signal comprising the steps of:

(a) subjecting said electrical communication signal to an adjustable inverse distortion operator to produce a channel distortion-compensated output signal; and

(b) estimating and updating one or more parameter values of said inverse distortion operator by processing said channel distortion-compensated output signal with at least one of

the output of a decision operator to which said channel distortion-compensated output signal is coupled, said decision operator being operative to produce an output data stream in accordance with prescribed decision criteria applied to said channel distortion-compensated output signal,

an undistorted version of a known signal pattern contained in said communication signal, and

prescribed statistics or other quantities of one or more system signals.

2. The method according to claim 1, wherein step (b) comprises updating parameter estimates of said inverse distortion operator by processing said channel distortion-compensated output signal and the output of said decision operator.

3. The method according to claim 2, wherein step (b) comprises generating estimates of parameters of said adjustable inverse distortion operator by combining said channel distortion-compensated output signal and the output of said decision operator to produce an error signal and coupling said error signal to a parameter estimate generator for said inverse distortion operator.

4. The method according to claim 1, wherein step (b) comprises updating parameter estimates of said adjustable inverse distortion operator by processing said channel distortion-compensated output signal and an undistorted version of a known signal pattern contained in said communication signal.

5. The method according to claim 4, wherein step (b) comprises updating parameter estimates of said adjustable inverse distortion operator by combining channel distortion-compensated output signal and an undistorted version of a known signal pattern contained in said communication sig-

nal to produce an error signal and coupling said error signal to a parameter estimate generator for said adjustable inverse distortion operator.

6. The method according to claim 1, wherein said known signal pattern comprises a frame synchronization pattern.

7. The method according to claim 1, wherein step (b) comprises subjecting said channel distortion-compensated output signal and said at least one of the output of said decision operator and said undistorted version of a known signal pattern contained in said communication signal to a prescribed synthesis operator to produce synthesized versions thereof, and processing said synthesized versions to update parameter estimates of said adjustable inverse distortion operator.

8. The method according to claim 1, wherein step (b) comprises updating parameter values of said inverse distortion operator in accordance with said prescribed statistics or other quantities of one or more system signals.

9. The method according to claim 8, wherein step (b) comprises updating parameter values of said inverse distortion operator in accordance with prescribed statistics or other quantities of said electrical communication signal.

10. A receiver apparatus for processing a communication signal that has been transported over a dispersive communication channel, and recovering therefrom an unknown information signal contained in said communication signal, wherein said communication signal is represented as an electrical communication signal, said receiver apparatus comprising:

an adjustable inverse distortion operator coupled to subject said electrical communication signal having a transfer function or characteristic that is effectively complementary to a distortion-introducing characteristic of said channel to produce a channel distortion-compensated output signal; and

a parameter estimate update mechanism, which is operative to update parameter estimates of said adjustable inverse distortion operator by processing said channel distortion-compensated output signal with at least one of

the output of a decision operator to which said channel distortion-compensated output signal is coupled, said decision operator being operative to produce an output data stream in accordance with prescribed decision criteria applied to said channel distortion-compensated output signal,

an undistorted version of a known signal pattern contained in said communication signal, and

prescribed statistics or other quantities of one or more system signals.

11. The receiver apparatus according to claim 10, wherein said parameter estimate update mechanism is operative to update estimates of parameters of said adjustable inverse distortion operator by processing said channel distortion-compensated output signal and the output of said decision operator.

12. The receiver apparatus according to claim 11, wherein said parameter estimate update mechanism is operative to generate estimates of parameters of said adjustable inverse distortion operator by combining said channel distortion-compensated output signal and the output of said decision operator to produce an error signal and coupling said error signal to a parameter estimate generator for said adjustable inverse distortion operator.

13. The receiver apparatus according to claim 10, wherein said parameter estimate update mechanism is operative to update parameter estimates of said adjustable inverse distortion operator by processing said channel distortion-compensated output signal and an undistorted version of a known signal pattern contained in said communication signal.

14. The receiver apparatus according to claim 13, wherein said parameter estimate update mechanism is operative to update parameter estimates of adjustable inverse distortion operator by combining channel distortion-compensated output signal and an undistorted version of a known signal pattern contained in said communication signal to produce an error signal and coupling said error signal to a parameter estimate generator for said adjustable inverse distortion operator.

15. The receiver apparatus according to claim 10, wherein said known signal pattern comprises a frame synchronization pattern.

16. The receiver apparatus according to claim 15, wherein said frame synchronization pattern comprises sequences of synchronous optical network (SONET) frame synchronization fields.

17. The receiver apparatus according to claim 10, wherein said parameter estimate update mechanism is operative to update parameter values of said inverse distortion operator in accordance with said prescribed statistics or other quantities of one or more system signals.

18. The receiver apparatus according to claim 7, wherein said parameter estimate update mechanism is operative to update parameter values of said inverse distortion operator in accordance with prescribed statistics or other quantities of said electrical communication signal.

19. A method of processing a communication signal, that has been transported over a dispersive communication channel, so as to recover an unknown information signal contained in said communication signal, comprising the steps of:

(a) coupling said communication signal to an adjustable inverse distortion operator which is operative to subject said communication signal to a transfer function that is effectively or approximately complementary to a distortion-introducing characteristic of said dispersive communication channel to produce a channel distortion-compensated output signal;

(b) performing a decision operation on said channel distortion-compensated output signal produced by said adjustable inverse distortion operator to produce a decision signal representative of said information signal;

(c) updating estimates of parameters of said adjustable inverse distortion operator by performing a prescribed combination of said channel distortion-compensated output signal with at least one of:

said decision signal,

an undistorted version of a known signal pattern contained in said communication signal, and

prescribed statistics or other quantities of one or more system signals.

20. The method according to claim 19, wherein step (c) comprises updating estimates of parameters of said adjustable inverse distortion operator by differentially combining said channel distortion-compensated output signal and said decision signal.

21. The method according to claim 20, wherein step (c) comprises generating estimates of parameters of said adjustable inverse distortion operator by combining said channel distortion-compensated output signal and said decision signal, and coupling said error signal to a parameter estimate generator for said adjustable inverse distortion operator.

22. The method according to claim 19, wherein step (c) comprises updating estimates of parameters of said adjustable inverse distortion operator by processing said channel distortion-compensated output signal and an undistorted version of a known signal pattern contained in said communication signal.

23. The method according to claim 22, wherein step (c) comprises estimating and updating parameter values of said adjustable inverse distortion operator by combining said channel distortion-compensated output signal and an undistorted version of a known signal pattern contained in said communication signal to produce an error signal and coupling said error signal to a parameter estimator for said adjustable inverse distortion operator.

24. The method according to claim 19, wherein step (c) comprises subjecting said channel distortion-compensated output signal and said at least one of the output of said decision signal and said undistorted version of a known signal pattern contained in said communication signal to a prescribed synthesis operator to produce synthesized versions thereof, and processing said synthesized versions to estimate and update parameter values of said adjustable inverse distortion operator.

25. The method according to claim 15, wherein step (e) comprises updating parameter values of said inverse distortion operator in accordance with said prescribed statistics or other quantities of one or more system signals.

26. The method according to claim 25, wherein step (e) comprises updating parameter values of said inverse distortion operator in accordance with prescribed statistics or other quantities of said electrical communication signal.

* * * * *

APPENDIX G

COPY OF HILLERY U.S. PATENT NO. 6,178,201 ("HILLERY")

US006178201B1

(12) **United States Patent**
Hillery

(10) **Patent No.: US 6,178,201 B1**
(45) **Date of Patent: Jan. 23, 2001**

(54) **CONTROLLING AN ADAPTIVE EQUALIZER IN A DEMODULATOR**

(75) Inventor: **William J. Hillery**, Santa Clara, CA (US)

(73) Assignee: **Agilent Technologies Inc.**, Palo Alto, CA (US)

(*) Notice: Under 35 U.S.C. 154(b), the term of this patent shall be extended for 0 days.

(21) Appl. No.: **09/038,656**

(22) Filed: **Mar. 11, 1998**

(51) **Int. Cl.**[7] **H03H 7/30**; H03H 7/40; H03K 5/159

(52) **U.S. Cl.** **375/232**; 708/323

(58) **Field of Search** ..... 375/232, 233, 375/230, 229; 348/607; 708/322, 323; 333/18, 28 R

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,418,817 * 5/1995 Richter 375/232
5,506,871 * 4/1996 Hwang et al. 375/230
5,517,213 * 5/1996 Bhatt et al. 375/232
5,809,074 * 9/1998 Werner et al. 375/233
5,835,532 * 11/1998 Strolle et al. 375/233
5,841,484 * 11/1998 Hulyalkar et al. 348/607

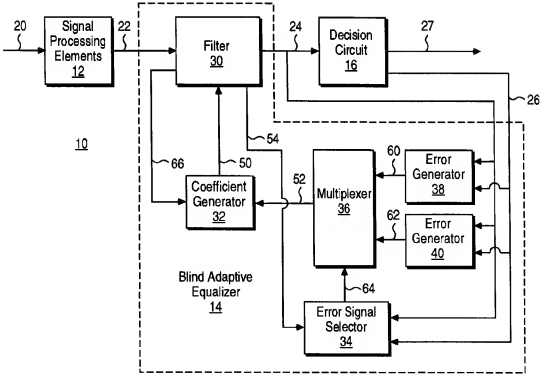
* cited by examiner

Primary Examiner—Stephen Chin
Assistant Examiner—Betsy L. Deppe

(57) **ABSTRACT**

An adaptive equalizer for a demodulator includes a filter that generates a filter output signal in response to an information signal according to a transfer function for the filter. The adaptive equalizer includes means for adjusting the transfer function in response to an indication of error in the filter output signal, and means for switching among a set of differing determinations of the indication of error. Switching among the differing determinations may be based upon a variety of indications of the progress of adaptation in the equalizer.

6 Claims, 3 Drawing Sheets



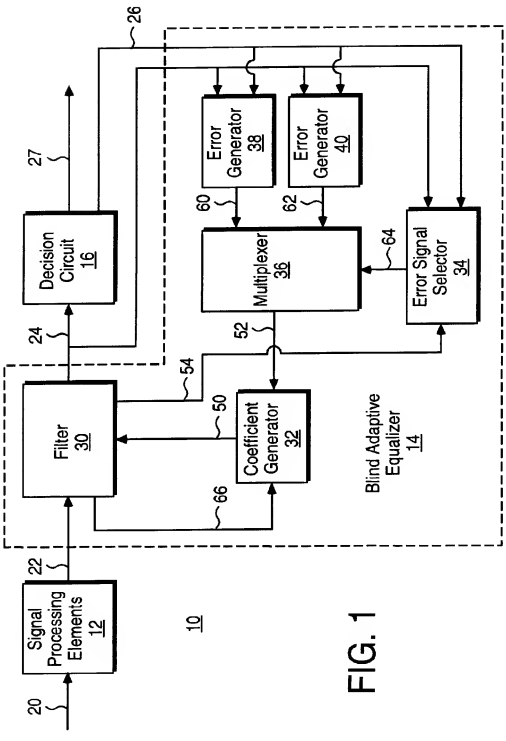

20
Signal Processing Elements 12
22
Filter 30
24
Decision Circuit 16
27
26
Error Generator 38
Error Generator 40
60
62
Multiplexer 36
64
Error Signal Selector 34
52
54
50
Coefficient Generator 32
66
Blind Adaptive Equalizer 14
10

FIG. 1

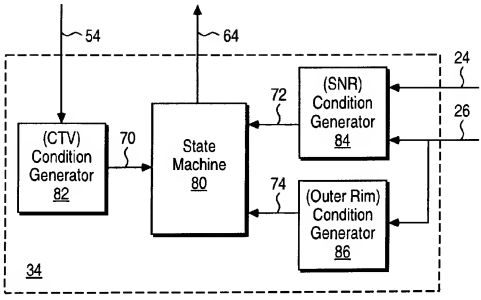

54
64
24
72
(SNR)
Condition
Generator
84
26
(CTV)
Condition
Generator
82
70
State
Machine
80
74
(Outer Rim)
Condition
Generator
86
34

FIG. 2

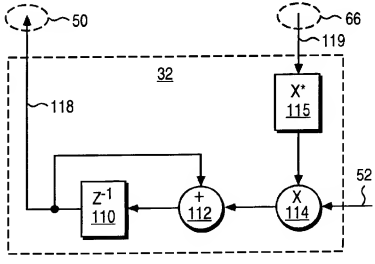
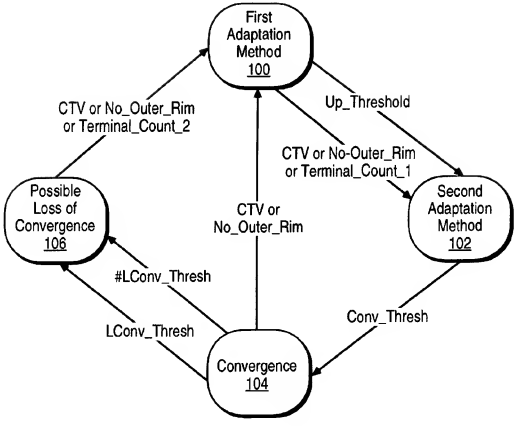

50
66
119
32
118
X*
115
52
Z-1
110
+
112
X
114

FIG. 4

FIG. 3


First Adaptation Method 100
Up_Threshold
CTV or No-Outer_Rim or Terminal_Count_1
Second Adaptation Method 102
Conv_Thresh
Convergence 104
LConv_Thresh
#LConv_Thresh
Possible Loss of Convergence 106
CTV or No_Outer_Rim or Terminal_Count_2
CTV or No_Outer_Rim

CONTROLLING AN ADAPTIVE EQUALIZER IN A DEMODULATOR

BACKGROUND OF THE INVENTION

1. Field of Invention

The present invention pertains to the field of demodulators. More particularly, this invention relates to controlling an adaptive equalizer in a demodulator.

2. Art Background

Communication systems commonly employ signal modulation to generate an information signal which is suitable for transmission via a physical communication path. For example, signal modulation is commonly employed in communication systems that transport information signals via transmission lines. Such communication systems include cable networks.

Such a communication system usually includes a transmitting station having a modulator that generates the information signal and a receiving station having a demodulator that extracts the information carried by the information signal. Typically, the transmitting and receiving stations are interconnected via one or more physical communication paths.

The communication paths in a typical communication system commonly include a variety of elements which introduce imperfections into the information signal. For example, transmission line connectors commonly cause signal reflections that distort the information signal. In addition, components such as signal amplifiers and filters may distort the information signal. Moreover, communication paths commonly have non-linear frequency and phase response which introduces further distortions into an information signal.

Prior demodulators may include an adaptive equalizer which is intended to compensate for the distortions which may be introduced into the information signal during transmission. A typical adaptive equalizer includes a filter and circuitry that continually adapts the filter according to a particular adaptation method. Prior adaptation methods are usually based upon a determination of an error measure for the adaptive equalizer. The adaptive equalizer is said to reach convergence when this error measure is small enough to yield a reliable output signal for the demodulator.

An adaptive equalizer may be classified as either blind or non-blind. A non-blind adaptive equalizer may be defined as an equalizer that adapts to a training sequence of symbols which is periodically carried in the information signal. A blind adaptive equalizer may be defined as an equalizer that adapts to a random symbol sequence in the information signal without the aid of a training sequence. It may in some systems be desirable to employ a blind adaptive equalizer. For example, a blind adaptive equalizer would eliminate the need for a training sequence which would ordinarily decrease the information throughput of a communication system.

Unfortunately, prior adaptation methods which may be useful for a blind adaptive equalizer commonly have difficulty in reaching a desirable point of convergence. For example, a common prior adaptation method in a blind adaptive equalizer employs a constant modulus algorithm (CMA) to recursively determine an error measure. Typically, the CMA adaptation method initially moves an equalizer toward convergence. The CMA adaptation method, however, usually has difficulty reaching a point of convergence that will yield a reliable output signal for the demodulator.

SUMMARY OF THE INVENTION

An adaptive equalizer for a demodulator is disclosed which switches among differing adaptation methods depending upon the progress of adaptation. The adaptive equalizer includes a filter that generates a filter output signal in response to an information signal according to a transfer function for the filter. The adaptive equalizer includes means for adjusting the transfer function in response to an indication of error in the filter output signal, and means for switching among a set of differing determinations of the indication of error. Switching among the differing determinations may be based upon a variety of indications of the progress of adaptation in the equalizer.

Other features and advantages of the present invention will be apparent from the detailed description that follows.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is described with respect to particular exemplary embodiments thereof and reference is accordingly made to the drawings in which:

FIG. **1** illustrates a demodulator which includes a blind adaptive equalizer that switches among differing adaptation methods;

FIG. **2** illustrates an error signal selector in one embodiment of the blind adaptive equalizer;

FIG. **3** is a diagram that shows the state transitions of a state machine in the error signal selector in one embodiment;

FIG. **4** illustrates elements of the coefficient generator.

DETAILED DESCRIPTION

FIG. **1** illustrates a demodulator **10** which includes a blind adaptive equalizer **14** that switches among differing adaptation methods. The demodulator **10** also includes a set of signal processing elements **12** and a decision circuit **16**. In one embodiment, the demodulator **10** is a 64-QAM demodulator which is characterized by a set of **64** constellation points each representing a symbol.

The signal processing elements **12** receive an information signal **20** which has been transported via a communication channel or a communication network. The communication channel or network may be embodied in one or more of a variety of physical communication paths including transmission line networks and broadcast communication channels. In one embodiment, the information signal **20** is received via an RF amplifier and associated circuitry which are coupled to a cable network.

The signal processing elements **12** perform a variety of signal processing functions to provide an input signal **22** for the equalizer **14**. In one embodiment, the signal processing elements **12** include an analog-to-digital converter, an automatic gain control, a matched filter, and a timing recovery function, as well as other signal processing elements.

A filter **30** generates a filter output signal **24** in response to the input signal **22**. In one embodiment, the filter **30** is a digital filter. In another embodiment, the filter **30** is a discrete time filter.

The relationship between the filter output signal **24** and the input signal **22** is defined by a transfer function associated with the filter **30**. The transfer function of the filter **30** is controllable and may be adjusted by the modification of a set of filter coefficients **50** which are generated by a coefficient generator **32**.

The decision circuit **16** generates an output signal **26** and an output signal **27** in response to the filter output signal **24**.

The output signal **27** is phase corrected and the output signal **26** is not phase corrected. The decision circuit **16** generates the phase-corrected output signal **27** by performing carrier recovery on the filter output signal **24** and selecting from among a set of predetermined constellation points which best match the symbols carried by the filter output signal **24**. The output signal **26** is the phase-corrected output signal **27** with phase correction removed. The predetermined constellation points are each represented by a complex number. Each predetermined constellation point has a real component and an imaginary component.

The coefficient generator **32** adapts the filter **30** to distortions or imperfections in the communication channel over which the information signal **20** was transported by continuously updating the filter coefficients **50** in response to an error signal **52**. The coefficient generator **32** updates the filter coefficients **50** according to the following:

$$w_n(k+1)=w_n(k)+e_{SEL}(k+1)y^*(k-n)$$

where k is time, e_{SEL} is the selected error signal **52**, n is the filter coefficient number (filter tap number), w_n is the filter coefficient for filter tap n, and $y(k-n)$ is the n^{th} entry in the delay line **66** at time k, and "*" indicates a complex conjugate.

The n filter taps from the filter **30** are provided to the coefficient generator **32** via a set of signal lines **66**. In addition to the equation set forth above, there are sign-based adaptation methods for generating updated filter coefficients which may be implemented in the coefficient generator **32**.

The error signal **52** is selected from among a set of differing error signals, N error signals in general. In one embodiment, N is 2 and the error signal **52** is selected as either an error signal **60** generated by an error generator **38** or an error signal **62** generated by an error generator **40**. The equalizer **14** includes a multiplexer **36** that selects either the error signal **60** or the error signal **62** in response to a control signal **64** generated by an error signal selector **34**.

The error generator **38** implements a preselected adaptation method which converges on a circle and is well suited for blind equalizers, i.e. to adaptations which are undertaken when no training symbol sequences are available in the information signal **20**. The error generator **38** determines an error in the filter output signal **24** and continuously updates the error signal **60** in response to this error determination. In one embodiment, the error generator **38** performs its error determination according to a constant modulus algorithm (CMA) which is as follows:

$$e_{CMA}(k+1)=\mu_{CMA}(R_2-|Z_k|^2)Z_k$$

where k is time, e_{CMA} is the error signal **60**, μ_{CMA} is a step size for the CMA algorithm, R_2 is a constellation dependent constant, and Z_p is the filter output signal **24**.

The error generator **40**, on the other hand, implements an adaptation method which is decision-directed toward the constellation points for the demodulator **10**. The error generator **40** determines an error in the filter output signal **24** and continuously updates the error signal **62** in response to this error determination. In one embodiment, the error generator **40** renders its error determination according a least mean-square (LMS) determination which is as follows:

$$e_{LMS}(k+1)=\mu_{LMS}(\hat{s}_k-Z_k)$$

where k is time, e_{LMS} is the error signal **62**, μ_{LMS} is a step size for the LMS algorithm, $\hat{s}_k$ is the output signal **26**, and Z_k, is the filter output signal **24**.

The error signals **60** and **62** are both indicators of the error between the output signal **26** and the filter output signal **24**. The process by which the adaptive equalizer **14** adjusts the transfer function of the digital filter **30** in a manner that reduces the error between the output signal **26** and the filter output signal **24** is called convergence. The adaptation method provided by the error generator **38** is selected for its efficiency in converging when relatively large differences exist between the output signal **26** and the filter output signal **24** as is common at the beginning of a convergence operation when no training symbols are available. On the other hand, the adaptation method provided by the error generator **40** is selected for its efficiency in accurately converging when relatively small differences exist between the output signal **26** and the filter output signal **24**.

The error signal selector **34** generates a set of status conditions and uses the status conditions to switch between the error signals **60–62**, where appropriate, to properly converge the adaptive equalizer **14** and compensate for the distortions in the input signal **22** which may have been caused by imperfections in the communication channel over which the information signal **22** was transported. The status conditions generally reflect the progress of a convergence operation being undertaken. The status conditions may reflect one or more thresholds in the signal-to-noise ratio of the demodulator **10**. The status conditions may also indicate whether a center tap of the filter **30** is outside of a predetermined range. The status conditions may also indicate the absence of outer rim constellation points. The error signal selector **34** uses the filter output signal **24**, the output signal **26**, and a center tap signal **54** from the digital filter **30** to determine the status conditions.

FIG. **2** illustrates the error signal selector **34** in one embodiment. In this embodiment, the error signal selector **34** includes a state machine **80** and a set of condition generators **82–86**. The state machine **80** switches the control signal **64** to select from among the error signals **60–62** in response to a set of control signals **70–74** which reflect the status conditions for a convergence operation. The control signals **70–74** are generated by the condition generators **82–86**.

The condition generator **82** detects false or improper convergence states which may occur where the center tap of the digital filter **30** is too large or too small or in the wrong position. In addition, the condition generator **82** detects when the equalizer **14** is not making progress toward convergence as may be indicated when the center tap of the digital filter **30** drifts outside a predetermined range. Any one or more of these conditions may be referred to as a center tap violation (CTV). The condition generator **82** uses the center tap signal **54** from the digital filter **30** to detect a CTV and asserts the control signal **70** to indicate a CTV to the state machine **80** if one of the following conditions is true.

$$(|Re\{C0\}|<Cmin) \text{ AND } (|Im\{C0\}|<Cmin)$$

$$(|Re\{C0\}|>Cmax) \text{ OR } (|Im\{C0\}|>Cmax)$$

where C0 is the complex value of the center tap signal **54**, Re{C0} is the real part of C0, Im{C0} is the imaginary part of C0, Cmin is the minimum allowed center tap value, and Cmax is the maximum allowed center tap value.

The condition generator **84** uses the filter output signal **24** and the output signal **26** to determine a signal-to-noise ratio (SNR) in the demodulator **10**. The signal level for the SNR determination is indicated by the signal power of the output

signal **26** and the noise level for the SNR determination is indicated by the signal power of the difference between the filter output signal **24** and the output signal **26**. The condition generator **84** compares the determined SNR to a set of SNR threshold values. These include an upper SNR threshold value, a convergence SNR threshold value, and a loss of convergence SNR threshold value.

The condition generator **84** indicates an Up_Threshold condition on the control signal **72** when the determined SNR is greater than the upper SNR threshold value. The Up_Threshold condition indicates that the adaptation method provided by the error generator **38**, CMA in one embodiment, has obtained sufficient convergence to enable a switch to the adaptation method provided by the error generator **40**, which in one embodiment is LMS.

The condition generator **84** indicates a Conv_Thresh condition on the control signal **72** when the determined SNR increases above the convergence SNR threshold value. This indicates that the adaptation method provided by the error generator **40** has obtained a high enough SNR in the demodulator **10** for a declaration of convergence by the equalizer **14**.

The condition generator **84** indicates an Lconv_Thresh condition on the control signal **72** after the equalizer **14** has converged if the determined SNR dips below the loss of convergence SNR threshold value. The Lconv_Thresh condition indicates a possible loss of convergence in the equalizer **14**. The condition generator **84** indicates an #Lconv_Thresh condition on the control signal **72** if the determined SNR rises above the loss of convergence SNR threshold value.

The condition generator **86** detects the lack of outer rim constellation points in the output signal **26**. For example, an outer rim constellation point for 64-QAM has a real component value of 3.5 in which case the condition generator **86** detects the lack of real component values of 3.5 in the output signal **26**. The condition generator **86** asserts the control signal **74** to indicate a No_Outer Rim condition if an outer rim value has not been detected within a predetermined number of previous symbols carried by the output signal **26**. In one embodiment, the condition generator **86** includes a counter which counts symbols and resets and reloads with a predetermined value whenever an outer rim value occurs. If this counter expires then the No_Outer_Rim condition is generated.

The No_Outer_Rim condition is used by the state machine **80** to prevent a false convergence of the equalizer **14**. If a false convergence occurs, the converged constellation is a smaller and noisier version of the actual constellation for the demodulator **10**. The lack of outer rim values is an indication of possible convergence to a false constellation.

FIG. **3** is a diagram that shows the state transitions of the state machine **80** in one embodiment. The states of the state machine **80** include a first adaptation method state **100**, a second adaptation method state **102**, a convergence state **104**, and a possible loss of convergence state **106**. Also shown are the status conditions as indicated on the control signals **70**–**74** that cause transitions among the states **100**–**106**.

The state machine **80** enters the first adaptation method state **100** at reset or initialization. While in the state **100**, the state machine **80** uses the control signal **64** to select the error signal **60** from the error generator **38** for use by the coefficient generator **32**. In one embodiment, this results in the use of the CMA adaptation method when updating the filter coefficients **50**. The state machine **80** remains in the first adaptation method state **100** so that the equalizer **14** uses the error signal **60** to converge until the condition generator **84** signals the Up_Threshold condition. The Up_Threshold condition causes the state machine **80** to transition to the second adaptation method at state **102**.

While in the state **102**, the state machine **80** uses the control signal **64** to select the error signal **62** from the error generator **40**. In one embodiment, this results in the use of the LMS adaptation method when updating the filter coefficients **50**. The state machine **80** transitions from the state **102** to the convergence state **104** once the condition generator **84** signals the Conv_Thresh condition. This indicates that the SNR in the demodulator **10** is high enough to declare the convergence of the equalizer **14**. On the other hand, the state machine **80** falls back from the state **102** to the state **100** if the condition generator **82** signals the CTV condition or if the condition generator **86** signals the No_Outer_Rim condition.

The state machine **80** includes a first counter which is loaded with a value indicating a maximum number of symbols for which the equalizer **14** will attempt to converge using the adaptation method of the error generator **40** in the state **102**. While in the state **102**, the state machine **80** decrements the first counter for each symbol. If the first counter expires while in the state **102** then the Terminal Count_1 condition is indicated and the state machine **80** falls back to the state **100** to restart the convergence process.

In the convergence state **104** the output signal **26** is considered to have good data and the state machine **80** continues to select the error signal **62** to maintain convergence. A CTV or a No Outer Rim indication in the convergence state **104** causes the state machine **80** to fall all the way back to the state **100** to restart the convergence process. While in the convergence state **104**, the Lconv_Thresh condition from the condition generator **84** causes the state machine **80** to transition to the possible loss of convergence state **106**.

While in the possible loss of convergence state **106**, the state machine **80** continues to select the error signal **62** for use by the coefficient generator **32**. A subsequent #Lconv_Thresh condition from the condition generator **84** restores the state machine **80** to the convergence state **104**. This maintains the use of the more finely tuned adaptation method of the error generator **40** in case the reduced SNR was caused by short term effects on the communication channel that transports the information signal **20**.

A CTV condition or a No_Outer_Rim condition causes the state machine **80** to fall out of the possible loss of convergence state **106** to the state **100** to restart the convergence process. The state machine **80** includes a second counter which is loaded with a value indicating a maximum number of symbols for which the equalizer **14** will attempt to re-attain convergence while in the possible loss of convergence state **106**. While in the possible loss of convergence state **106**, the state machine **80** decrements the second counter for each symbol. If the second counter expires as indicated by the Terminal_Count_2 condition then the state machine **80** transitions to the state **100** to restart the convergence process.

FIG. **4** illustrates elements of the coefficient generator **32**. The coefficient generator **32** includes a delay element, an adder, a complex conjugate block, and a multiplier for each of the filter coefficients **50**. For generating a filter coefficient **118**, the coefficient generator **32** includes a delay element **110** such as a register or flip-flop, an adder **112**, a multiplier **114** and a complex conjugate block **115**.

The complex conjugate block **115** generates a conjugate of a corresponding filter tap **119**. The multiplier **114** multi-

plies the output of the complex conjugate block **115** by the selected error signal **52**. The delay element **110** stores a previous coefficient w(k) and the adder **112** generates an updated coefficient w(k+1) by adding the previous coefficient w(k) to the output of the multiplier **114**. The output of the delay element **110** provides the filter coefficient **118** for the corresponding filter tap **119**.

The filter **30** may be implemented in a variety of arrangements including a simple feed-forward filter and a filter having feed-forward and feed-back elements. In one embodiment, the filter **30** includes a feed-forward filter from the input signal **22** and a feed-back filter from the output signal **26**. The outputs of the feed-forward and feedback filter are summed to provide the filter output signal **24**. The center tap signal **54** is the last tap in the feed-forward filter.

The foregoing detailed description of the present invention is provided for the purposes of illustration and is not intended to be exhaustive or to limit the invention to the precise embodiment disclosed. Accordingly, the scope of the present invention is defined by the appended claims.

What is claimed is:

1. An adaptive equalizer for a demodulator, comprising:

filter that generates a filter output signal in response to an information signal according to a set of coefficients for the filter;

means for continuously updating the coefficients in response to an indication of error in the filter output signal;

means for switching among a set of differing determinations of the indication of error while continuously updating the coefficients wherein the means for switching switches among the differing determinations in response to a status indication in the demodulator and wherein the status indication is based upon a determination of whether a center tap signal from the filter exceeds a set of predetermined boundaries.

2. An adaptive equalizer for a demodulator, comprising:

filter that generates a filter output signal in response to an information signal according to a set of coefficients for the filter;

means for continuously updating the coefficients in response to an indication of error in the filter output signal;

means for switching among a set of differing determinations of the indication of error while continuously updating the coefficients wherein the means for switching switches among the differing determinations in response to a status indication in the demodulator and wherein the status indication is based upon a detection of a lack of an outer rim constellation point in an output signal of the demodulator.

3. An adaptive equalizer for a demodulator, comprising:

filter that generates a filter output signal in response to an information signal according to a transfer function for the filter;

a set of error generators each of which provides a differing determination of an indication of error in the filter output signal;

circuit that switches among the error generators to provide a selected indication of error;

coefficient generator that continuously generates updated filter coefficients that adjust the transfer function of the filter in response to the selected indication of error;

wherein the circuit that switches comprises a multiplexer that selectively couples the indications determined by the error generators to the coefficient generator and a status condition generator that generates a status condition indicating progress in adapting the filter and a state machine that causes the multiplexer to switch among the indications from the error generators in response to the status condition and wherein the status condition indicates whether a center tap signal from the filter exceeds a set of predetermined boundaries.

4. An adaptive equalizer for a demodulator, comprising:

filter that generates a filter output signal in response to an information signal according to a transfer function for the filter;

a set of error generators each of which provides a differing determination of an indication of error in the filter output signal;

circuit that switches among the error generators to provide a selected indication of error;

coefficient generator that continuously generates updated filter coefficients that adjust the transfer function of the filter in response to the selected indication of error;

wherein the circuit that switches comprises a multiplexer that selectively couples the indications determined by the error generators to the coefficient generator and a status condition generator that generates a status condition indicating progress in adapting the filter and a state machine that causes the multiplexer to switch among the indications from the error generators in response to the status condition and wherein the status condition indicates a lack of an outer rim constellation point in an output signal of the demodulator.

5. A method for adapting an equalizer in a demodulator, comprising the steps of:

generating a filter output signal in response to an information signal according to a transfer function having a set of coefficients;

generating a set of indications of error in the filter output signal, each indication based upon a differing determination of error in the filter output signal;

generating a selected indication of error by switching among the indications of error;

continuously updating the coefficients in response to the selected indication of error;

wherein the step of switching among the indications comprises the steps of generating a status condition indicating progress in adapting the filter output signal and switching among the indications in response to the status condition and wherein the step of generating a status condition includes the step of determining whether a center tap signal associated with the transfer function exceeds a set of predetermined boundaries.

6. A method for adapting an equalizer in a demodulator, comprising the steps of:

generating a filter output signal in response to an information signal according to a transfer function having a set of coefficients;

generating a set of indications of error in the filter output signal, each indication based upon a differing determination of error in the filter output signal;

generating a selected indication of error by switching among the indications of error;

continuously updating the coefficients in response to the selected indication of error;

wherein the step of switching among the indications comprises the steps of generating a status condition

indicating progress in adapting the filter output signal and switching among the indications in response to the status condition and wherein the step of generating a status condition includes the step of detecting a lack of an outer rim constellation point in an output signal of the demodulator.

* * * * *

APPENDIX H

COPY OF LU U.S. PATENT NO. 6,275,836 ("LU")

US006275836B1

(12) **United States Patent**
Lu

(10) **Patent No.: US 6,275,836 B1**
(45) **Date of Patent: Aug. 14, 2001**

(54) **INTERPOLATION FILTER AND METHOD FOR SWITCHING BETWEEN INTEGER AND FRACTIONAL INTERPOLATION RATES**

(75) Inventor: **Jinghui Lu**, Austin, TX (US)

(73) Assignee: **Oak Technology, Inc.**, Sunnyvale, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/094,528**

(22) Filed: **Jun. 12, 1998**

(51) **Int. Cl.**[7] **G06F 7/10**; G06F 7/17
(52) **U.S. Cl.** **708/313**; 708/316
(58) **Field of Search** 708/300, 313, 708/103, 316; 341/61

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,079,734		1/1992	Riley .	
5,455,782		10/1995	Young et al. .	
5,786,778	*	7/1998	Adams et al.	341/61
5,916,301	*	6/1999	Rothacher	708/313
5,963,153	*	10/1999	Rosefield et al.	341/61

OTHER PUBLICATIONS

Franca et al., *Design of Analog–Digital VLSI Circuits for Telecommunications and Signal Processing*, Second Edition, Prentice Hall 1994, pp. 251–288.

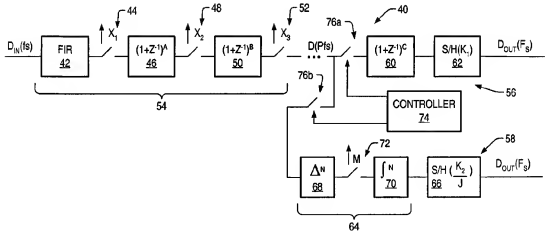
* cited by examiner

Primary Examiner—Chuong Dinh Ngo
(74) *Attorney, Agent, or Firm*—Kevin L. Daffer; Conley, Rose & Tayon P.C.

(57) **ABSTRACT**

A voice or data transmission system and specifically an interpolation filter used within the transmission system is provided for producing either fractional or integer interpolation ratios. The digital signal resulting from the interpolation filter has a relatively high signal to noise ratio whenever fractional interpolation is needed. The interpolation filter includes multiple stages coupled in series, and an integer interpolation branch switched in parallel with a fractional interpolation branch. A controller determines whether the integer or fractional interpolation ratio is needed based on maintaining a fixed oversampling data rate from the interpolation filter given a changing incoming sampling rate. If the incoming sampling rate should require fractional interpolation, then a branch implementing fractional interpolation ratio is used in lieu of the integer interpolation ratio. A comb filter is preferably introduced within the fractional interpolation branch to attenuate imaging tones within the baseband of interest. An interpolation rate change switch used by the comb filter beneficially moves the imaging tones further away from the baseband so that minimum imaging noise is introduced within the baseband by the fractional oversampling ratio.

20 Claims, 3 Drawing Sheets



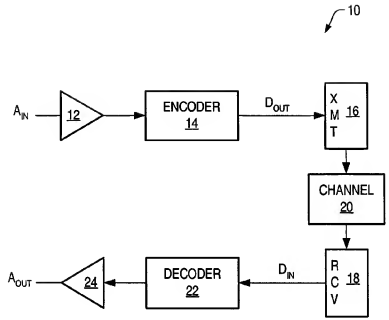

10
A_{IN}
12
ENCODER
14
D_{OUT}
X
M
T
16
CHANNEL
20
R
C
V
18
D_{IN}
DECODER
22
24
A_{OUT}

FIG. 1

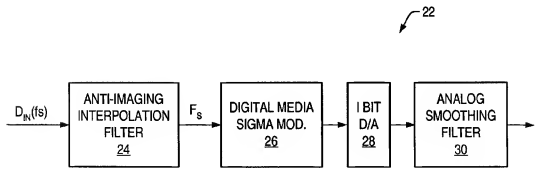

22
D_{IN}(fs)
ANTI-IMAGING INTERPOLATION FILTER
24
F_s
DIGITAL MEDIA SIGMA MOD.
26
I BIT D/A
28
ANALOG SMOOTHING FILTER
30

FIG. 2

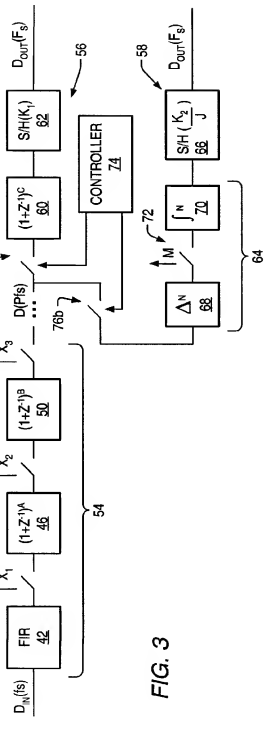

$D_{IN}(fs)$
FIR 42
44
X_1
$(1+Z^{-1})^4$ 46
48
X_2
$(1+Z^{-1})^8$ 50
52
X_3
54
$D(P_{fs})$
76a
76b
40
$(1+Z^{-1})^C$ 60
S/H(K_1) 62
$D_{OUT}(F_s)$
56
CONTROLLER 74
72
M
Δ^N 68
$\int^N$ 70
64
58
S/H ($\frac{K_2}{J}$) 66
$D_{OUT}(F_s)$

FIG. 3

fs
2fs
(X1-X2-X3)fs
Pfs
2Pfs
$F_s=K_1Pfs$

FIG. 4

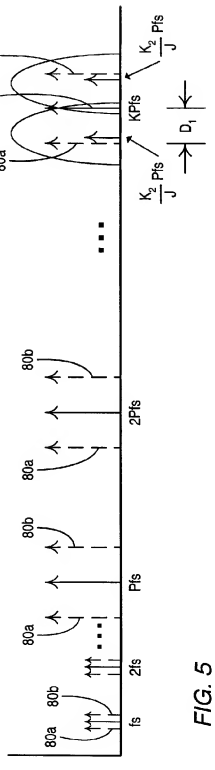

80a
80b
fs
2fs
Pfs
2Pfs
MPfs
2MPfs
KPfs
82
92
94a
94b
90a
90b
K_2 Pfs
J
K_2 MPfs
J
K_2MPfs
$F_s = \frac{K_2}{J}$ MPfs
D_1
D_2

FIG. 5

FIG. 6

INTERPOLATION FILTER AND METHOD FOR SWITCHING BETWEEN INTEGER AND FRACTIONAL INTERPOLATION RATES

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates to a telecommunications system which can perform digital-to-analog ("D/A") conversion, or to an interpolation filter which can switch between integer and fractional interpolation rates to accommodate a fixed oversampling output rate for differing digital signal input frequencies.

2. Description of the Related Art

Mixed signal integrated circuits are generally well known as those which operate on both analog and digital signals. Such an integrated circuit may include analog-to-digital ("A/D"), D/A and digital signal processing (DSP) functions on a single monolithic substrate.

There are numerous types of mixed signal integrated circuits. One popular example involves a CODEC. Generally speaking, a CODEC may involve manipulating both analog and digital signals useful in the telecommunications field. An analog input signal is generally converted to a digital format, where multiple real-time operations can be readily performed on the digital signal before transmitting the digital signal onto a transmission medium. The CODEC can further include a mechanism for receiving digital signals from the transmission medium and converting those received signals to an analog format. Thus, encoding into a digital format and decoding from the digital format allows DSP operations to be performed using rapid digital operations. Operations such as multiplication, summing and delay can occur in succession on the digital signals before and after they are transmitted to enhance their transmissivity. These elemental operations an be performed many times for each sample of the incoming signal.

FIG. 1 is a block diagram of a CODEC **10**, according to one conventional form. CODEC **10** includes an amplifier **12** having gains sufficient to forward an analog input signal A_{IN} to an encoder **14**. Encoder **14** may include many operational blocks which convert the analog input signal to a digital bit stream with minimum noise induced within the passband of interest. A popular A/D converter includes oversampled converters, or delta-sigma converters. Essentially, a delta-sigma converter digitizes an analog signal at a very high sampling rate (i.e., oversampling) to perform a noise-shaping function. Digital filtering after noise-shaping allows the delta-sigma converter to achieve a higher effective resolution than the analog input signal. Decimation can be used to reduce the oversampling data rate back to the original "Nyquist" rate.

The Nyquist data rate of the digital bit stream D_{OUT} can be forwarded to a transmitter **16** as shown. Transmitter **16** as well as receiver **18** are situated near ports of CODEC **10** to transmit and receive respective digital signals through a channel **20** operably linked to CODEC **10**. Similar to transmitter **16**, receiver **18** may include amplification and/or error correction circuitry which operates solely within the digital domain. Resulting from receiver **18** is a digital bit stream D_{IN} forwarded to a decoder **22**. The analog output from decoder **22** can thereafter be amplified by amplifier **24** and presented back as an analog output signal A_{OUT}. Similar to encoder **14**, decoder **22** includes a delta-sigma converter and a digital filter. However, instead of using the delta-sigma converter as an A/D converter followed by a digital decimation filter, decoder **22** is configured with an interpolation filter coupled to receive the digital input D_{IN} prior to performing D/A conversion which uses, in part, a delta-sigma modulator.

FIG. **2** illustrates in more detail an exemplary set of blocks which may be used to form decoder **22**. Several bit lines of a digital word can be fed as D_{IN} to an interpolation filter **24**. The data rate of each digital word is shown at frequency fs. Interpolation filter **24** includes a sampling rate conversion switch which increases the sampling rate of the incoming bit stream D_{IN} from fs to a higher sampling rate Fs. The technique of sample rate conversion used in interpolation or decimation is generally well known. Included with sample rate conversion may be a filter transfer function carried out by operations performed by the DSP portion of the mixed signal integrated circuit. An execution unit which performs a summing, delay and/or multiplication operation can be used to implement any filter transfer function.

Delta-sigma modulator **26** within decoder **22** proves useful in noise-shaping the high data rate bit streams and may also serve to reduce a multi-bit digital word to a single sample width of one bit. Unlike a delta-sigma modulator within an A/D block, modulator **26** exclusively performs digital operations. The transfer function is implemented in the digital domain possibly with an infinite impulse response ("IIR") filter. The transfer function filtering operation performs the same modulator function as the A/D unit, where the in-band noise is suppressed but induces higher frequency quantization noise.

Output from delta-sigma modulator **26** can be fed as a single bit data stream transitioning at a relatively high rate Fs into a one bit D/A converter **28**, and the output of converter **28** is filtered in the analog domain by smoothing filter **30**. Smoothing filter **30** helps remove the shaped quantization noise induced by modulator **26**, and rejects any images which result above the output Nyquist rate fs.

A mixed signal integrated circuit is shown in FIGS. **1** and **2** to involve not only a mechanism to form a digital signal, but also recover analog information from that digital signal using an interpolation filter **24**, followed by a delta-sigma modulator **26**. The interpolation filter **24** thereby forms part of a transmission system and, more particularly, part of a D/A converter (or decoder). The multi-bit word line of interpolation filter **24** can receive a bit stream of varying data rate (i.e., frequency) depending on the telecommunications application. For example, the transmission system and particularly the CODEC can be called upon to send either data or voice information. This implies that the incoming data stream D_{IN} can have a wide variety of frequencies fs depending on the transmitter/receiver constraints as well as the transmission medium. A variety of frequencies fs can also arise depending on the transmission system used and/or the transmission application desired. For example, modem applications involving data transmissions may utilize different frequencies fs than audio applications involving voice. However, to maintain compatibility with an existing (fixed) modulator **26** and D/A converters **28**, the data rate Fs should remain consistent when changing from one input frequency fs to the next. This will enhance applicability of an existing D/A converter to many different input frequencies.

A need, therefore, exists for having an interpolation filter which can accommodate varying baseband sampling frequencies fs, and can interpolate those varying frequencies to a fixed oversampling frequency Fs for use by a delta-sigma modulator operating at a fixed sampling rate. The desirable interpolation filter would thereafter enhance applicability of existing D/A converters to many different input frequencies.

In order to accommodate a wide variety of input frequencies fs, the desired interpolation filter must not only involve a programmable interpolation rate, but also must be capable of selecting between fractional and integer interpolation rates on-the-fly.

SUMMARY OF THE INVENTION

The problems outlined above are in large part solved by an improved interpolation filter hereof. The interpolation filter can accommodate an incoming signal transitioning at varying sample rates fs. If the sample rate should change, the interpolation filter will modify its interpolation rate to maintain a constant and fixed oversampling output data rate Fs. The oversampling output can be maintained even if the interpolation rate involves switching between an integer and fractional interpolation ratio. In this manner, the interpolation filter used within the D/A converter can accommodate a transmission signal forwarded at virtually any incoming digital signal frequency less than half the sample rate.

The interpolation ratio supported by the present interpolation filter is programmable as either an integer number or a number having both integer and fractional values. The interpolation filter also does not require decimation at any stage within the interpolation filter. Thus, the programmable interpolation ratio does not produce a data rate which extends above the oversampled output Fs of the interpolation filter. Accordingly, throughout the present interpolation filter, an integer and/or fractional interpolation rate change occurs exclusively. The interpolation filter thereby avoids costly polyphase filters and unduly high intermediate frequencies associated with polyphase filters. The programmable interpolation rate involves possibly many stages of interpolation which increase the data rate directly to the oversampled output value without involving any intermediate or interim decimation stages. It has been determined that the present interpolation filter can achieve greater than 94 dB signal-to-noise performance.

According to one embodiment, the interpolation filter involves two interpolation branches coupled in parallel between a common input conductor and a common output conductor. One branch is used to accommodate integer interpolation, and the other branch accommodates fractional interpolation. Integer interpolation preferably involves a cosine filter (i.e., cosine transfer function) coupled in series with an interpolation sample and hold switch. The series-connected cosine filter and the sample and hold switch are connected between the input conductor and the output conductor. The second branch preferably involves a comb filter coupled in series with a sample and hold interpolation switch between the input conductor and the output conductor. Thus, the two branches can be considered coupled in parallel and a set of switches can be used to select between the branches depending on the digital bit stream received on the input conductor. The bit stream forwarded to the input conductor can be pre-conditioned possibly by upstream interpolation switches and associated filters.

Broadly speaking, a transmission system is set forth including a transmitter and a receiver. The receiver is coupled to a decoder, and the decoder is configured to receive a digital signal from the receiver. The digital signal transitions at a frequency which can vary depending on the application to which the transmission system is subjected. The decoder includes a first interpolation filter contained within a first branch and a second interpolation filter contained within a second branch. The first interpolation filter receives the digital signal during a first time duration and produces a first digital output signal having a data rate increased from the digital signal frequency by an integer number (i.e., K_1). The second interpolation filter receives the digital signal during a second time duration and produces a second digital output signal which transitions at a data rate increased by a fractional number (i.e., K_2/J) from that of the digital signal frequency. Additionally, the second digital output transitions at a data rate increased by the product of an integer number M and the fractional number K_2/J from that of the digital signal.

According to one embodiment, the first interpolation filter within the first branch is coupled in parallel with the second interpolation filter within the second branch. The first interpolation filter may involve a cosine filter followed by a sample and hold interpolation switch having a rate change factor K_1. The second interpolation filter may involve a cosine interpolation filter having a rate change factor M, followed by a sample and hold switch having a rate change factor K_2/J. Rate change factor K_1 may or may not be equal to K_2, depending on the incoming sample rate fs.

The digital signal frequency fs of the incoming digital signal can vary significantly depending on, for example, whether voice or data is being transmitted. Regardless of fs, the sample rate output from the first and second branches during the respective first and second time durations remains the same. That is, Fs remains fixed since the first and second interpolation filters can accommodate dissimilar sample rate increases.

The cosine filter within the first branch may receive a digital signal transitioning at a first frequency during a first time duration and the comb interpolation filter within the second branch may receive the digital signal transitioning at a second frequency during a second time duration. Thus, the interpolation filter may further include a controller for connecting the digital signal to either the cosine interpolation filter or the comb interpolation filter depending on whether the digital signal is transitioning at the first frequency or the second frequency. The controller may therefore contain circuitry which senses the data rate of the incoming digital signal. Based on that sample rate, the controller then either (i) changes the amount by which the integer sample rate increasing switch modifies the sample rate if integer modification can be used, (ii) switches from the first branch to the second branch if fractional sample rate modification is needed, (iii) changes the fractional sample rate modification factor in the second branch if the incoming digital signal sample rate changes and still requires fractional sample rate modification, or (iv) switches back to the first branch if sample rate modification can be performed by an integer value rather than a fractional value. Of course, there may be numerous other scenarios by which the controller selects between the first branch and the second branch and modifies the sample rate change factors within those branches.

In addition to the transmission system, the D/A converter or the interpolation filter hereof, a method may be presented for inducing an interpolation rate change to a digital bit stream depending on the status of that bit stream. More specifically, the bit stream can be sampled at a first sample rate, and the first sample rate can be increased by an integer value to an oversampled rate Fs. The digital bit stream can then be sampled at a second sample rate dissimilar from the first sample rate which may therefore involve increasing the second sample rate by a fractional value to the oversampled rate Fs. Receiving the first and second sample rates involves switching the digital bit stream from an integer interpolation filter to a fractional interpolation filter when the first sample rate of the incoming signal changes to the second sample rate.

BRIEF DESCRIPTION OF THE DRAWINGS

Other objects and advantages of the invention will become apparent upon reading the following detailed description and upon reference to the accompanying drawings in which:

FIG. **1** is block diagram example of a conventional mixed signal system;

FIG. **2** is a block diagram example of the decoder illustrated in FIG. **1**;

FIG. **3** is a symbolic representation of a switchable, multi-staged interpolation filter according to one embodiment;

FIG. **4** is a frequency spectrum of an oversampled output frequency Fs produced from the interpolation filter of FIG. **3** in relation to a digital input signal sampled at frequency fs;

FIG. **5** is a frequency spectrum indicating a problem of an aliasing band encroaching upon the oversampled output frequency Fs whenever the interpolation rate includes a fraction; and

FIG. **6** is a frequency spectrum resulting from a comb interpolation filter introduced before the fractional sample and hold switch to reduce the alising band magnitude within Fs and to increase the relative separation between Fs and the aliasing band.

While the invention may be modified and have alternative forms, specific embodiments thereof are shown by way of example in the drawings and will herein be described in detail. It should be understood, however, that the drawings and detailed description thereto are not intended to limit the invention to the particular form disclosed, but on the contrary, the intention is to cover all modifications, equivalents and alternatives falling within the spirit and scope of the present invention as defined by the appended claims.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Turning to FIG. **3**, a block diagram of a multi-staged interpolation filter **40** is shown. Filter **40** can be implemented possibly within a D/A converter, or within a decoder of a digital signal transmission system. Interpolation filter **40** combines interpolation sample rate change switches with digital filtering transfer functions (i.e., DSP summing and delay operations) across multiple stages. This results in an increase in the computational efficiency if done correctly. There are numerous digital filters which can be implemented with interpolation. A popular filter includes the finite impulse response ("FIR") filter **42**, often denoted as a moving weighted average filter. An advantage of FIR filters is that filter outputs need only be computed at the lower sample rate (prior to interpolation rate change switch **44**) thereby achieving considerable efficiency in the computational process. The FIR filter **42** is only one example of a filter which receives the digital input signal D_{fs}, sampled at rate fs. Whatever filter transfer function is used, it is desired that the filtering characteristic demonstrate a fairly sharp reduction in amplitudes beyond a defined frequency value. For this reason, FIR filters may be employed as an early stage of interpolation filter **40**.

Subsequent to interpolation rate change switch **44**, a cosine filter **46** may be used. Cosine filter **46** is shown having a transfer function of $(1+z^{-1})^A$. The term A refers to the number of orders or terms linked in series from the output of sample rate change switch **44** and sample rate change switch **48**. A single order cosine filter involves a delay element which receives an input digital bit stream and sums the delayed output with the digital bit stream input to produce a summed output. A general discussion of FIR filters and cosine filters, and the use of those filters in multiple stages implemented with multiple rates is set forth in, for example, U.S. Pat. Nos. 5,079,734 and 5,455,782, and further in an article to Franca, et al. entitled "Design of Analog-Digital VLSI Circuits for Telecommunications and Signal Processing," (Prentice Hall, 2nd Ed., 1994) pp. 251–288 herein incorporated by reference.

The transfer function of a cosine filter is described above as a pair of operations: a delay element and a summing node. Thus, the $1+z^{-1}$ is a term embodied as a hardware element or a microcode sequence. A suitable cosine operation of a single term can be carried out by one add and one delay element connected together in feedforward arrangement. If connected in feedforward with a negative add at the sum terminal, the term is said to represent a differentiator. If connected in feedback with a positive add at the sum terminal, the term represents an integrator. The combination of differentiators and integrators are useful in describing a sample and hold switch which essentially operates as a comb filter having differentiator and integrator components. As will be described herein below, a comb filter is desirably used in a second branch of interpolation filter **40** for the benefit of performing fractional sample rate modification. A comb filter is therefore dissimilar from a cosine filter which is carried out by feedforward with a positive add at the sum terminal. Of course, integrators, differentiators and cosine transfer terms may be separated by up sampling rate change switches in order to effectuate sample rate conversion among the various stages of those shown in FIG. **3**.

Coupled to the output of interpolation switch **48** may be another cosine filter having possibly the same number of terms A or a dissimilar number of terms B used to represent the order of cosine filter **50**. Another interpolation switch **52** may be connected subsequent to filter **50**. The combination of switches **44**, **48**, and **52** with filters **42**, **46**, and **50** produce an overall, multi-staged interpolation filter front-end portion **54** of interpolation filter **40**. The various filter transfer functions or terms can be changed, and the sequence or values of interpolation switches can be modified provided the incoming digital signal is filtered from noise contained on that signal. The amount of sample rate increase can also change and thus the values for the staged interpolation switches **44**, **48**, and **52** are represented as X_1, X_2, and X_3, respectively. It is noted herein below that the cumulative product of $X_1*X_2*X_3$ is equal to P. Regardless of the value of P and the mount or composition of filtering at front-end portion **54**, a digital signal D sampled at ate Pfs is presented to one of two branches **56** or **58** depending on the variable, incoming ample rate Pfs relative to the desired, fixed output sample rate Fs.

First branch **56** preferably includes a cosine filter **60** followed by a rate change switch **62**. Cosine filter **60** may have the same number of terms or orders as filter **46** or filter **50**, or can have an altogether different number of orders labeled C. According to one example, C=2. It is noted that an increase in the number of orders will present a sharper cutoff at the passband edges. The cosine filter will accept signals only within a defined frequency range and will suffice to limit (i.e., attenuate) aliasing signals presented near the sidelobes of the cosine-filtered passband. Interpolation switch **62** provides an integer increase K_1 in sample rate. The Fourier transform of sample signals at the output of switch **62** has periodic images centered around all multiples of the sampling rate according to well known sampling theory. Insertion of zero-valued samples, or held

samples, by sample rate conversion switch **62** does not alter the frequency domain description of the periodic images which remain centered around all multiples of the sampling rate. Cosine filter **60** serves to substantially eliminate images which fall far away from, or become aliased upon, the base band signal. This can be readily performed since the base band signal is centered around integer multiples of the sampling rate Pfs.

The second branch **58** includes a comb filter **64** and an interpolation rate change switch **66**. Comb filter **64** can be represented as a differentiator **68** separated from an integrator **70** by an interpolation switch **72**. Rate change switch **66** serves to increase the sample rate by a fractional value, or a value which includes a fractional number. The fractional or integer combined with fractional number is represented as K_2/J.

A controller **74** is used to control interpolation switches **76***a* and **76***b*. If the incoming data rate is of a certain magnitude that requires a fractional sample rate increase, then second branch **58** is used. However, if the incoming data rate merely requires an integer sample increase, then first branch **56** is used. Controller **74** determines which branch is to be used based on the incoming data sampling rate and selects a desired switch **76***a* or **76***b* depending on whether integer or fractional interpolation is needed. Controller **74** includes any sensing circuit, and a circuit responsive to the sensed signal for applying a selection voltage value to switch **76**, possibly configured as a transistor.

FIG. **4** is a graph of frequency vs. amplitude and, more specifically, represents a frequency spectrum and response magnitudes prior to front-end portion **54** and subsequent to front-end portion **54**. Accordingly, the frequency spectrum represents the incoming sampling data rate fs after undergoing an initial sampling by an interpolation rate change switch to present multiple periodic images centered around multiples P of the sampling rate fs. The images continue for multiples of Pfs as a result of the sample and hold switch **62**. Accordingly, FIG. **4** illustrates the frequency spectrum of a digital input signal D_{IN} forwarded through front-end portion **54** and first branch **56** to present an output signal D_{OUT} sampled at frequency Fs. The output sampling rate $Fs=K_1*Pfs$, and therefore is represented as an integer multiple of the original sampling rates fs. The images repeat at integer intervals of the incoming sampling rate fs, where out-of-baseband aliasing signals can be readily removed by cosine filters **46**, **50**, and **60**, if desired.

FIG. **4** therefore represents the controller selecting the first branch for integer interpolation rate change. However, if the incoming data rate changes and fractional interpolation rate change is needed to maintain Fs fixed to that which was previously produced, then periodic images **80***a* and **80***b* centered around multiples of sampling rate fs (i.e., centered about integer KPfs) will encroach into the fractional baseband region of $PfsK_2/J$ shown in FIG. **5**. The baseband of interest is less than an integer value away from integer multiples of sampling rate fs. It is noted that KP are integers, and the product of K_2P is an integer. The imaging or aliasing bands **80***a* and **80***b* are found to be near KPfs and, unfortunately, within the baseband of $PfsK_2/J$. The fractional output sampling rate Fs of $PfsK_2/J$ will therefore receive the aliasing tones unless a comb filter is implemented before the fractional interpolation switch.

FIG. **5** illustrates the frequency spectrum absent a comb filter. Thus, the comb filter **64** shown in FIG. **3** provides at least two distinct benefits. Comb filter of order N will present sharp attenuation of any aliasing tones that fall near the edge of the baseband. The greater the value of N, the greater the attenuation near the side lobes of the baseband. The comb filter can therefore be tuned by increasing the order of the comb filter and thereby suppressing or attenuating aliasing or imaging signals imaged from KPfs but relatively near the output sampling data rate.

A further advantage of the comb filter is the introduction of an additional sample rate conversion factor M. If M is sufficiently large, then the separation of the aliasing tones from KMPfs will be much larger than the separation between the imaging tones and a smaller sampling rate KPfs.

FIG. **6** illustrates the benefits of introducing a comb filter to solve the problem shown in FIG **5**. Specifically, the additional interpolation factor **M** multiplied with the pre-existing interpolation factor P will cause relative movement of the imaging tones **90***a* and **90***b* attributed to the multiple integer number KMPfs. Imaging tones **90***a* and **90***b* are represented as a proportion of the central sampling rate MPfs, similar to the ratio of imaging tones **80***a* and **80***b* relative to the central sampling rate Pfs, also shown in FIG. **6**. FIG. **5** shows the relative difference D_1 between the central, multiple integer sampling rate **82** and the imaging tones **80**. The additional separation (or difference D_2) between multiple integer sampling rate **92** and tones **90***a* and **90***b* shown in FIG. **6** is partially brought about by the introduction of sample rate interpolation factor M used to move the rather large aliasing tone **90** outside the baseband **94**.

Although not restricted to a comb filter, a comb filter can be implemented whenever fractional interpolation rate change is needed. A filter such as a comb filter beneficially improves the signal to noise ratio to a value exceeding 94 db. The transfer filter functions within the first and second branches between sample rate Pfs to Fs are as follows:

$$\text{Integer Ratio: } H(z)=(1+z^{-1})^{N}*((1-z^{-KJ})/(1-z^{-1}))$$

$$\text{Fractional Ratio: } H_c(z)=((1-z^{-M})/(1-z^{-1}))^{N*}((1-z^{-KJ2^{n}})/(1-z^{-1}))$$

where $z=e^{-j\omega}$.

The upper and lower branches **56** and **58** are switched onto the data path depending on whether integer or fractional interpolation ratios are needed. Merely as an example, if Fs must remain at 6.144 MHz, then a change in fs from 48.0 KHz to 44.1 KHz requires the controller to switch from the integer interpolation ratio to the fractional interpolation ratio (i.e., switch from the first branch to the second branch). The following represents examples various incoming data rates fs which may be used, the branch used, and the corresponding exemplary values for K_1, K_2, P, M, and J:

TABLE I.

fs	K_P (integer)	MPK_2/J (fractional)	Fs
48.0 KHz	128	not applicable	6.144 MHz
44.1 KHz	not applicable	418/3	6.144 MHz
22.05 KHz	not applicable	836/3	6.144 MHz

It would be appreciated by those skilled in the art having the benefit of this disclosure that the transmission system, D/A converter, and the interpolation filter with switchable branches, or method thereof, is believed capable of applications in which noise can be reduced within a specific frequency range. Furthermore, it is also to be understood that the invention shown and described is to be taken as presently preferred embodiments. Various modifications and changes may be made to the design structure to optimize the

term count and the overall filter composition as would be obvious to a person skilled in the art having the benefit of this disclosure. The specification and drawings are therefore to be regarded in an illustrative rather than a restrictive sense.

What is claimed is:

1. A transmission system, comprising:

a receiver; and

a decoder operably coupled to receive a digital signal from the receiver, wherein the digital signal operably transitions at a digital signal frequency fs, and wherein the decoder includes:

a first interpolation filter adapted to receive a digital input signal derived from the digital signal and produce a first digital output signal during a first time duration, wherein the first digital output signal transitions at a data rate increased by an integer number K_1 from that of the digital input signal;

a second interpolation filter adapted to receive the digital input signal and produce a second digital output signal during a second time duration dissimilar from the first time duration, wherein the second digital output signal transitions at a data rate increased by a fractional number from that of the digital input signal; and

a control circuit adapted to direct the digital input signal to the first interpolation filter during the first time duration and to the second interpolation filter during the second time duration.

2. The transmission system as recited in claim **1**, wherein the decoder further includes a modulator coupled to the first and second interpolation filters for receiving the first digital output signal during the first time duration and thereafter receiving the second digital output signal during the second time duration.

3. The transmission system as recited in claim **1**, wherein the control circuit selects between the first and second interpolation filters to maintain a fixed data rate whenever the digital signal frequency changes.

4. The transmission system as recited in claim **1**, wherein a frequency of the digital input signal is equal to Pfs, wherein P is an integer.

5. The transmission system as recited in claim **4**, wherein the fractional number comprises a quotient of an integer K_2 and a value J such that the product of Pfs and K_2/J during the second time duration is equal to the product of Pfs and K_1 during the first time duration.

6. The transmission system as recited in claim **5**, wherein the fractional number further comprises an integer **M** multiplied by the quotient K_2/J such that the product of Pfs and MK_2/J during the second time duration is equal to the product of Pfs and K_1 during the first time duration.

7. The transmission system as recited in claim **5**, wherein a first value of fs occurs during the first time duration and a second value of fs occurs during the second time duration.

8. An interpolation filter, comprising:

a first interpolation filter coupled along a first signal path between an input conductor and an output conductor for performing an integer rate change during a first time duration;

a second interpolation filter coupled along a second signal path between the input conductor and the output conductor for performing a fractional rate change during a second time duration dissimilar from the first time duration, wherein the first signal path is connected in parallel with the second signal path;

a switch configured to connect the input conductor to the first interpolation filter during the first time duration and the second interpolation filter during the second time duration; and

a third interpolation filter coupled between the input conductor and the switch.

9. The interpolation filter as recited in claim **8**, wherein the output conductor is adapted to receive a digital output signal of a fixed data rate during both the first and second time durations.

10. The interpolation filter as recited in claim **8**, wherein the output conductor is adapted to receive a digital output signal having a data rate during the first time duration that is equal to a data rate during the second time duration.

11. The interpolation filter as recited in claim **8**, wherein said third interpolation filter comprises multiple stages.

12. An interpolation filter, comprising:

a cosine interpolation filter coupled in series with an integer sample rate increasing switch for receiving a digital signal transitioning at a first frequency during a first time duration; and

a comb interpolation filter coupled in series with a fractional sample rate increasing switch for receiving the digital signal transitioning at a second frequency during a second time duration.

13. The interpolation filter as recited in claim **12**, further including a controller for connecting the digital signal to either the cosine interpolation filter or the comb filter depending on whether the digital signal is transitioning at the first frequency or the second frequency.

14. The interpolation filter as recited in claim **12**, wherein the digital signal results from at least one interpolating sample rate increasing switch.

15. The interpolation filter as recited in claim **12**, wherein the frequency of the digital signal transitioning during the first time duration times a magnitude of the integer sample rate increasing switch is equal to an oversampling frequency Fs, and the oversampling frequency Fs is equal in magnitude to the digital signal transitioning during the second time duration times a magnitude of the fractional sample rate increasing switch.

16. A method for inducing an interpolating rate change to a digital bit stream, comprising:

receiving a digital bit stream sampled at a first sample rate;

increasing the first sample rate by an integer value to an oversampled rate Fs using a first interpolation filter;

receiving the digital bit stream sampled at a second sample rate dissimilar from the first sample rate; and

increasing the second sample rate by a fractional value to the oversampled rate Fs using a first interpolation filter.

17. The method as recited in claim **16**, wherein the oversampled rate Fs remains fixed regardless of whether the digital bit stream is sampled at the first sample rate or the second sample rate.

18. The method as recited in claim **16**, wherein said increasing the second sample rate comprises switching the digital bit stream from an integer interpolation filter to a fractional interpolation filter when the first sample rate changes to the second sample rate.

19. The method as recited in claim **16**, wherein said increasing the second sample rate comprises comb filtering the digital bit stream prior to feeding the digital bit stream to a fractional sample rate increasing switch.

20. The method as recited in claim **16**, wherein said increasing the first sample rate comprises cosine filtering the digital bit stream prior to feeding the digital bit stream to the integer sample rate increasing switch.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 6,275,836 B1
DATED : August 14, 2001
INVENTOR(S) : Lu

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 10, claim 16,
Line 49, after the phrase "the oversampled rate Fs using a" please delete the word "first" and substitute therefor -- second --.

Signed and Sealed this

Ninth Day of April, 2002

Attest:

Attesting Officer

JAMES E. ROGAN
Director of the United States Patent and Trademark Office

APPENDIX I
PEDERSEN ET AL. U.S. PATENT APPLICATION
PUBLICATION NO. 2006/0114979 ("PEDERSEN")

US 20060114979A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2006/0114979 A1

Pedersen et al. (43) Pub. Date: Jun. 1, 2006

(54) FULLY PARAMETRIC EQUALIZER

(76) Inventors: **Kim Rishoj Pedersen**, Iga (DK); **Knud Bank Christensen**, Ryomgard (DK)

Correspondence Address:
CANTOR COLBURN, LLP
55 GRIFFIN ROAD SOUTH
BLOOMFIELD, CT 06002

(21) Appl. No.: **10/538,593**

(22) PCT Filed: **Dec. 9, 2002**

(86) PCT No.: **PCT/DK02/00831**

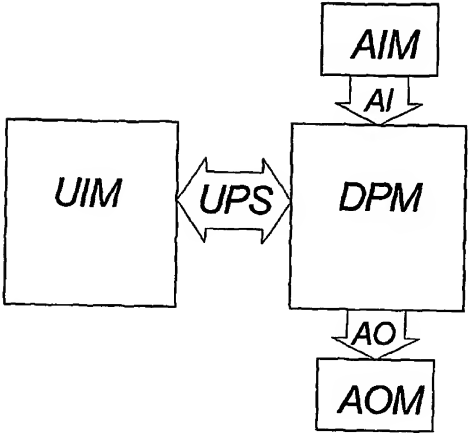
Publication Classification

(51) **Int. Cl.**
H03H 7/30 (2006.01)

(52) **U.S. Cl.** **375/229**

(57) **ABSTRACT**

The invention relates to a parametric equalizer comprising filtering means (FM), user interface means (UIM), audio signal input means and audio signal output means, said filtering means comprising at least one filter block (FIB) said user interface means (UIM) facilitating adjustment of corner frequency (fc), Shape (Q) and gain (G), said user interface means (UIM) comprising a further adjustment parameter (SYM), said user interface means being mapped by means of coefficient adjustment algorithms (CAD) into filter coefficient settings (FCP) of the at least one filter block (FIB), which when established reflects the adjustment of the user interface means (UIM) said further adjustment parameter (SYM) providing a filter coefficient setting (FCS) comprising a combined adjustment of at least one zero frequency, pole frequency, zero Q and pole Q of at least one filter block.



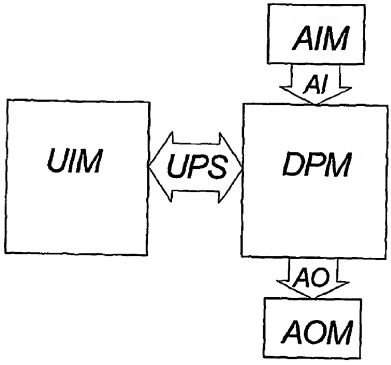


Fig.1

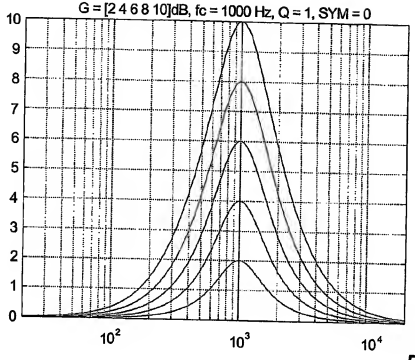

G = [2 4 6 8 10]dB, fc = 1000 Hz, Q = 1, SYM = 0
10
9
8
7
6
5
4
3
2
1
0
10^2
10^3
10^4

Fig. 2a

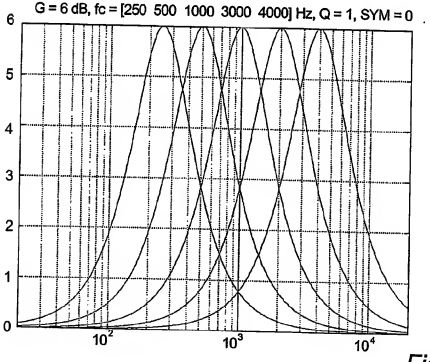

G = 6 dB, fc = [250 500 1000 3000 4000] Hz, Q = 1, SYM = 0
6
5
4
3
2
1
0
10^2
10^3
10^4

Fig. 2b

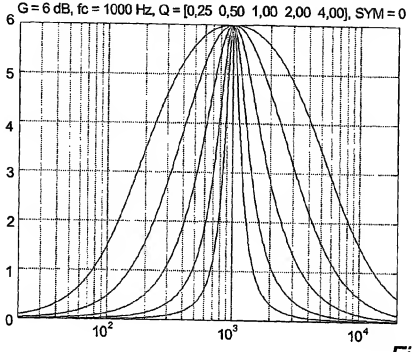

G = 6 dB, fc = 1000 Hz, Q = [0,25 0,50 1,00 2,00 4,00], SYM = 0
6
5
4
3
2
1
0
10^2
10^3
10^4

Fig. 2c

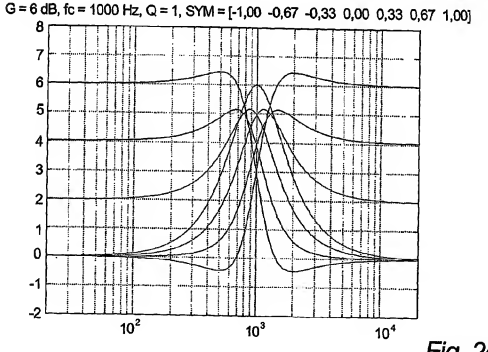

G = 6 dB, fc = 1000 Hz, Q = 1, SYM = [-1,00 -0,67 -0,33 0,00 0,33 0,67 1,00]
8
7
6
5
4
3
2
1
0
-1
-2
10^2
10^3
10^4

Fig. 2d

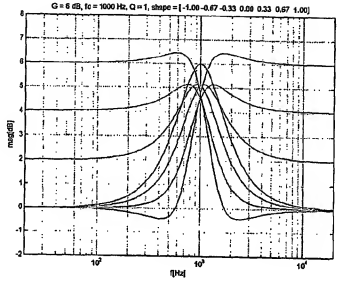


Fig. 3a

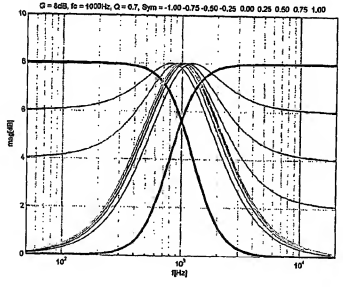


Fig. 3b

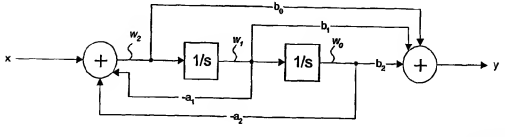


Fig. 4

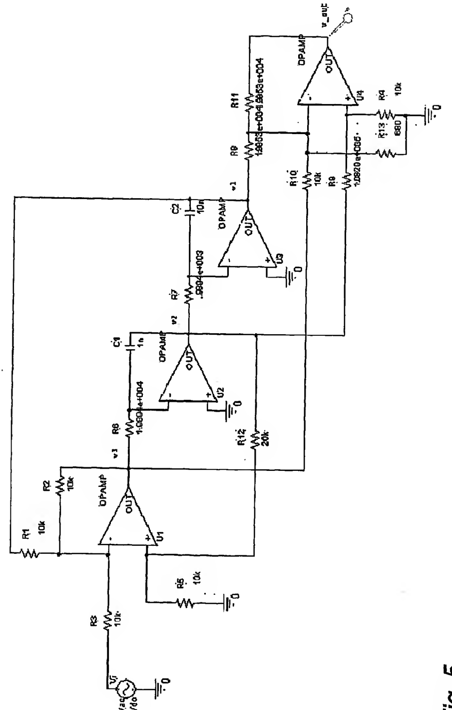

OPAMP
OUT
U1
U2
U3
U4
R1
10k
R2
10k
R3
10k
R5
10k
R6
R7
R9
R10
10k
R11
R12
20k
R13
680
R4
10k
C1
1n
C2
10n
v1
v2
v3
v_out

Fig. 5

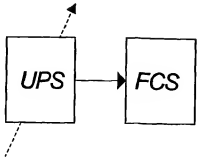


Fig.6a

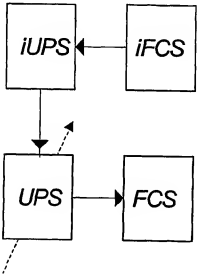


Fig.6b

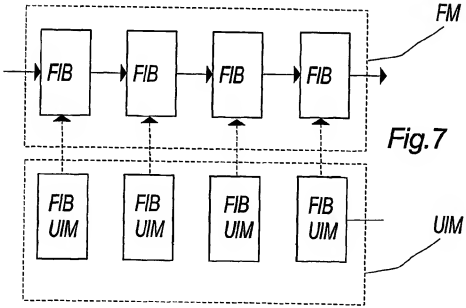


Fig.7

FULLY PARAMETRIC EQUALIZER

FIELD OF THE INVENTION

[0001] The present invention relates to a parametric equalizer according to claim 1.

BACKGROUND OF THE INVENTION

[0002] Today's equalizers can be divided into graphical and parametrical equalizers. Both types can be implemented in analog or digital signal processing technology. This invention deals with parametric equalizers in analog as well as digital implementations.

[0003] Generally a graphical equalizer benefits of the fact that the complete audio spectrum may be divided into several fixed regions with levels controlled in an intuitive way by a user of the equalizer. A significant problem of the graphical equalizer is, however, that the equalizer is quite inflexible and provides very little possibility of accurate control by a user due to the fact that the user is typically restricted to utilization of the pre-defined bands. In practice, such problem would only be solved by the use of even more than thirty bands. Such a device would typically be a very expensive device simply because of sheer duplication of circuitry. Much of the circuitry will be wasted when dealing with several types of equalizing tasks because such types of tasks typically only involve adjustment of two or three bands while others should be left unaffected by the filtering.

[0004] An equalizer with adjustable frequency has therefore been provided for the purpose of optimizing the use of signal processing circuitry. Such an equalizer is referred to as a parametric equalizer and has upon introduction in 1972 found wide use especially in professional or semi-professional contexts.

[0005] Basically a parametric equalizer features very few control parameters, which, on the other hand may control the curve response with very high resolution.

[0006] Typical control parameters are gain, center frequency and Q. Moreover, some parametric equalizers provide three different filter types, a low shelf filter, a bell-shaped filter and a high shelf filter.

[0007] A parametric equalizer can produce a very sharp notch, as a graphic equalizer, and maintain the shape over several decades or bands. A parametric equalizer may, contrary to most applicable graphical equalizers produce a magnitude response boost or attenuation at any frequency and may therefore match the average desired sound filtering characteristic somewhat better than graphic equalizers.

[0008] An application of a parametric equalizer may for example be suppression of low frequency microphone noise.

[0009] A typical parametrical equalizer may comprise a number of filter blocks, which may be cascaded for the purpose of obtaining one desired combined transfer function of the cascaded filters.

[0010] Presently, each block is typically one of three types: Low-shelf, parametric (bell-shaped) and high-shelf filter. The low/high shelf filters have three parameters each: Gain G, corner frequency f and slope (or Q), and the parametric filters have three similar parameters: Gain G, center frequency f_c and bandwidth BW (or Q).

[0011] A problem of the prior art parametric equalizers is that the freedom of operation is somewhat limited and that the full use of the parametric equalizers for certain common desired filter characteristics requires several cascaded filters, thereby increasing the complexity of the complete system and evidently, increasing the costs involved, due to the fact that several units must be applied for the purpose of obtaining the single desired characteristics.

[0012] It is the object of the invention to provide an equalizer featuring the above-mentioned advantages of prior art parametric equalizers while ameliorating the above-mentioned disadvantages of the prior art.

SUMMARY OF THE INVENTION

[0013] The invention relates to a parametric equalizer comprising

filtering means (FM), user interface means (UIM), audio signal input means and audio signal output means,

said filtering means comprising at least one filter block (FIB)

said user interface means (UIM) comprising means for adjustment of parameters: corner frequency (fc), shape (Q) and gain (G),

said parametric equalizer comprising means for establishing a variable magnitude response symmetry of said at least one filter block (FIB).

[0014] According to the invention, it should be noted that the adjustment parameter referred to as a non-trivial parameter and mentioned as "gain" above refers to the sign characteristic of the log magnitude response of the applied filter, i.e. whether the filter defines a boost or an attenuation at the corner frequency.

[0015] According to the invention, non-trivial degrees of freedom are the degrees of freedom left, when the overall gain is disregarded or simply handled as a product of the overall gains of the individual filter blocks. In other words, the complete number of degrees of freedom, when dealing with for example a biquadratic filter block, is five, that is one trivial degree of freedom being the overall gain of the filter block and four non-trivial degrees of freedom. This understanding of degrees of freedom is further explained in the detailed description.

[0016] Elsewhere, the parameter global gain or overall gain refers to a trivial parameter corresponding to the linear volume setting of the applied filter block or group of filter blocks.

[0017] According to the invention, the further adjustment parameter, exemplified by the symmetry parameter, facilitates the possibility of adjustment of the symmetry of the filter magnitude response, both by providing conventional obtainable filter types, such as low-shelf, bell-shaped and high shelf and mixtures or intermediates (with respect to gain symmetry) thereof.

[0018] Such intermediate filter would according to one embodiment of the invention comprise a continuos interval of curve shape defined by variation of the symmetry parameters according to the invention. According to a preferred embodiment of the invention, the symmetry may be varied between low frequency gain boost/attenuation anti-symme-

try via center frequency symmetry (e.g. bell-shaped) to high frequency boost/attenuation anti-symmetry.

[0019] According to a preferred embodiment of the invention, a continues interval (may of course be established as a high resolution set of discrete filters in the digital world) of filter shapes having magnitude response symmetry varying from one sign of asymmetry to the opposite sign of asymmetry. Preferably, the available continuous number of filter symmetries should comprise the symmetrical instance of the filter design corresponding to the bell-shape.

[0020] It should be noted that the gain, Q and fc preferably may be adjusted at every available setting of the Symmetry parameter.

[0021] An example of available equalizer filters according to an embodiment of the invention is a filter featuring adjustable asymmetrical over-/undershoot of the filter magnitude response at the selected corner frequency, gain and Q.

[0022] According to the invention, the new adjustment parameter may be referred to as the symmetry parameter. The variable symmetry parameter should not be confused with the shape aspects referring to prior art parametric equalizers' variable Q.

[0023] According to the invention, the improved control of the equalizer may in fact surprisingly be obtained "free of charge" due to the fact that the improved control may in fact be obtained by the use of conventional filter types, such as biquadratic filter blocks, only now utilizing all four non-trivial degrees of freedom simultaneously.

[0024] As it has been appreciated the filtering structure of a parametric equalizer may be regarded as relatively simple at least in the sense that the huge number of processing blocks of for example a graphic equalizer may be avoided.

[0025] According to the present invention, this advantage has been maintained while adding significant adjustment features to the user.

[0026] According to a particular user-friendly embodiment of the invention, the adjustment parameter may be practically "dimmed" for the purpose of emulating a conventional parametric equalizer. In this way, a user feeling uncomfortable with the adjustment opportunities provided according to the invention may simply convert the equalizer into a conventional and familiar sound-processing device.

[0027] In an embodiment of the invention, the user interface means UIM comprises a further symmetry adjustment parameter SYM for establishing a variable symmetry of the magnitude response of said at least one filter block FIB.

said user interface means is mapped by means of coefficient adjustment algorithms into filter coefficient settings FCS of the at least one filter block FIB, which when established reflects the adjustment of the user interface means UIM

said further adjustment parameter SYM provides a filter coefficient setting FCS comprising a combined adjustment of at least one zero frequency, pole frequency, zero Q and pole Q of the magnitude response at least one filter block.

[0028] In an embodiment of the invention said user control means facilitates adjustment of corner frequency, fc, shape, Q, gain and symmetry, SYM.

[0029] In an embodiment of the invention said filter coefficient settings FCS comprise digital coefficients.

[0030] In an embodiment of the invention said filter coefficient settings FCS comprises analogue values established by means of adjustable analogue filter components of said at least one filtering means.

[0031] In an embodiment of the invention said filtering means comprises less than twenty individually adjustable filter blocks FIB, preferably less that ten and most preferably less than six.

[0032] It should be noted that the filter blocks of a filtering means, e.g. a parametric equalizer preferably should be individually adjustable, thereby facilitating the cascading and adjusting of some or all the filter blocks of the parametric equalizer.

[0033] In an embodiment of the invention at least one of said filtering blocks comprises biquad filters (biquad: biquadratic).

[0034] In an embodiment of the invention said parametric equalizer comprises at least one, preferably at least three cascaded biquadratic filters.

[0035] In an embodimcnt of the invention said filtering means is analogously implemented.

[0036] In an embodiment of the invention said filtering means is digitally implemented.

[0037] In an embodiment of the invention said filtering means comprises gain compensation means adapted for compensation of alteration of the filtering block gain invoked by a changed setting of the further adjustment parameter.

[0038] In an embodiment of the invention said filtering means comprises corner frequency compensation means adapted for compensation of alteration of the corner frequency of the filtering block invoked by a changed setting of the further adjustment parameter SYM.

[0039] In an embodiment of the invention said further adjustment parameter is adapted for providing an adjustment of both the asymmetry around the corner frequency of at least one filter block FIB and the asymmetry around the half gain of the at least one filter block over at least a part of the frequency range of the filter block.

[0040] In an embodiment of the invention said user interface provides at least four different asymmetries of filter setting for at least a part of the frequency range.

[0041] In an embodiment of the invention said further adjustment parameter SYM enables the user to gradually transform the filter block FIB between a low-shelf filter characteristic and a high-shelf.

[0042] It should be noted that several other desirable asymmetries than the well known low-shelf and high shelf equalizer filters may define the endpoints of the available asymmetrics. Even though it is highly preferred that the available symmetries (or rather asymmetries) are defined within an interval of asymmetries in order to facilitate the user to grasp the available modifications, discrete, non-continuous sets of filter characteristics may be offered.

[0043] In an embodiment of the invention said further adjustment parameter (SYM) enables the user to gradually transform the filter block (FIB) from a low-shelf into a bell-shape and further into a high-shelf, thus defining at least one more than the three standard filter types.

[0044] In an embodiment of the invention the number of said adjustment parameters correspond to the number of degrees of freedom of the at least one filter block.

[0045] In an embodiment of the invention the number of said adjustment parameters is four times the number of non-trivial degrees of freedom of at least one biquad filter block.

[0046] In an embodiment of the invention the number of non-trivial degrees of freedom of each of a number of cascaded filter blocks is at least four.

[0047] A further degree of freedom may be a global gain setting, which may be associated to each filter block or may be shared as a global gain setting shared by all the connectable. typically cascadable, filter blocks.

[0048] In an embodiment of the invention the symmetry parameter may be set by means of the user interface to at least four different values, preferably a continues interval of values in the analog or digital embodiment.

[0049] In an embodiment of the invention the adjustment parameters are converted into filter coefficient settings (FCS) triggered by the setting of the adjustment parameters by the user.

[0050] According to the invention, the filter coefficient settings may be established "on the fly" triggered by the setting of the adjustment parameters by a user. In this way, memory may be saved.

[0051] In an embodiment of the invention the conversion of adjustment parameters into filter coefficient settings is invertible.

[0052] In an embodiment of the invention the given filter coefficient settings may be converted into corresponding adjustment parameters.

[0053] According to the invention, an initially applied filter may be presented to the user in corresponding parametric equalizer parameter settings. Moreover, the filter may then be tuned by the parametric equalizer according to an embodiment of the invention.

[0054] In an embodiment of the invention a method of adjusting the filter coefficients of the filter of a parametric equalizer comprises the step of availing user adjustment of all the degrees of freedom of the transfer function or a block of the transfer function of the filter.

[0055] In an embodiment of the invention said availing of user adjustment comprises the steps of adjusting four degrees of freedom per filter block.

[0056] In an embodiment of the invention adjustment of the filter coefficients is implemented in a parametric equalizer according to any of claims **1**-**23**.

[0057] Moreover, the invention relates to a method of adjusting the filter coefficients of the filter of a parametric equalizer comprising the step of availing user adjustment of all the degrees of freedom of the transfer function or a block of the transfer function of the filter.

THE DRAWINGS

[0058] The invention is described in the following with reference to the drawings of non-limiting examples, of which

[0059] **FIG. 1** illustrates the principle components applied according to an embodiment of the invention,

[0060] **FIG. 2** illustrates the filter characteristics according to an embodiment of the invention,

[0061] **FIG. 3***a* illustrates a frequency compensated embodiment of the invention,

[0062] **FIG. 3***b* illustrates a gain compensated embodiment of the invention,

[0063] **FIG. 4** illustrates a block diagram of analog state-variable filter

[0064] **FIG. 5** illustrates a circuit diagram of single analog biquad filter according to an embodiment of the invention,

[0065] **FIGS. 6***a* and **6***b* illustrate the principle of the invertability obtained according to an embodiment of the invention, and

[0066] **FIG. 7** illustrates a cascade of filter block in an embodiment of the invention, and

DETAILED DESCRIPTION

[0067] **FIG. 1** illustrates the principle components of a parametric equalizer according to an embodiment of the invention.

[0068] The main hardware components comprise User Interface Means UIM, Data Processing Means DPM, Audio Input Means AIM and Audio Output Means AOM.

[0069] The User Interface Means UIM is adapted for, under the control of a user, establishment of the adjustable parameters controlling the data processing of the Data Processing Means DPM by means of User Parameter Settings UPS controlling the Data Processing Means DPM.

[0070] The Data Processing Means DPM comprises suitable data processing hardware and associated circuitry, including memory, clock generators, etc. The Data Processing Means receives Audio Input signals AI provided by the Audio Input Means AIM and outputs Audio Output AO signals to the Audio Output Means AOM.

[0071] The Audio Input signals may comprise digital or analog signals. In case of analog signals, the Audio Input Means AIM or the Data Processing Means DPM should preferably comprise the necessary A/D-converters. In case of digital Audio Input AI signals, Audio Input Means AIM or the Data Processing Means DPM should preferably comprise suitable input means.

[0072] The User Interface Means UIM comprises suitable adjustments means adapted for manual use. The adjustment means may preferably comprise conventional buttons/kiobs/sliders/etc. and associated display means (not shown) or for example be controlled by a computer implemented interface (not shown) comprising the conventional user input means, such as keyboard and/or mouse and monitor.

[0073] Turning now to the theoretical background of the invention.

[0074] Classic parametric EQ functions comprise adjustment parameters: Low shelf, parametric and high shelf with varying G,fc and Q

[0075] As mentioned above these filters are typically implemented as biquadratic blocks (analog case shown here):

$$H(s)=\frac{b_0s^2+b_1s+b_2}{a_0s^2+a_1s+a_2}=g_{overall}\frac{s^2+\frac{\omega_z}{Q_z}s+\omega_z^2}{s^2+\frac{\omega_p}{Q_p}s+\omega_p^2}$$

$$\text{where } g_{overall}=\frac{b_0}{a_0},\ \omega_z=\sqrt{\frac{b_2}{b_0}},\ Q_z=\frac{\sqrt{b_0b_2}}{b_1},$$

$$\omega_p=\sqrt{\frac{a_2}{a_0}},\ Q_p=\frac{\sqrt{a_0a_2}}{a_1}$$

[0076] It can be seen that H(s) has 5 degrees of freedom: The overall gain of the individual filter block, which is trivial—equivalent to a volume control-, and 4 non-trivial ones, namely the resonance frequencies and Qs of the numerator and denominator respectively. So each of the standard filter types use only 3 out of 4 degrees of freedom, leaving one degree of freedom un-utilized; shelves let $Q_p=Q_z$ while parametric bell filter let $\omega_p=\omega_z$. To put it another way the 3 standard filter types (low-shelf, parametric and high-shelf) are but samplings along a 4th parameter axis that has so far been hidden from the user.

The Symmetry Parameter

[0077] The new parameter will be referred to as the Symmetry parameter, and according to an embodiment of the invention it is defined so that the three traditional filter types correspond to Symmetry=−1, 0 and 1 respectively. A first implementation of the new parameter goes like this (Algorithm 1):

[0078] Given user parameters G in dB, f_c in Hz, Q and symmetry:

$$g=10^{\frac{G}{20}}$$

$$\omega=2\pi f_c$$

$$\omega_z=\omega\cdot g^{-\frac{symmetry}{4}}$$

$$\omega_p=\frac{\omega^2}{\omega_z}$$

$$Q_z=Q\cdot g^{|symmetry|-1}$$

$$g_{correction}=\begin{cases}1 \text{ if } symmetry\le 0\\ \left(\frac{\omega_p}{\omega_z}\right)^2 \text{ otherwise}\end{cases}$$

$$H(s)=g_{correction}\frac{s^2+\frac{\omega_z}{Q_z}s+\omega_z^2}{s^2+\frac{\omega_p}{Q_p}s+\omega_p^2}$$

$$\text{if } G<0\text{: } H(s)=H(s)^{-1}$$

[0079] It should be noted that other definitions or adaptation of the symmetry parameter may be applied according to the invention.

[0080] Response examples of a fully parametric EQ, one parameter variation at a time, is illustrated in **FIG.** 2*a*-2*d*.

[0081] **FIG.** 2*a* illustrates a response of the parametric equalizer with variable gain, fixed fc=1000 Hz, fixed Q=1 and fixed Symmetry=0.

[0082] **FIG.** 2*b* illustrates a response of the parametric equalizer with fixed gain=6 dB, variable corner frequency fc, fixed Q=1 and fixed Symmetry=0.

[0083] **FIG.** 2*c* illustrates a response of the parametric equalizer with fixed gain=6 dB, fixed corner frequency fc=1000 Hz, variable Q and fixed Symmetry=0.

[0084] **FIG.** 2*d* illustrates a response of the parametric equalizer with fixed gain=6 dB, fixed corner frequency fc=1000 Hz, fixed Q=1 and variable Symmetry.

[0085] It should be noted that the illustrated obtainable curve forms incorporate both the traditional available settings and the complete range of the fourth parameter, Symmetry.

[0086] This is pinpointed in **FIG.** 2*d*, where the obtained filter characteristic itself is advantageous and where the filter may be obtained by advantageous and simple control.

[0087] In the embodiment of the invention illustrated in **FIG.** 2*d*, the SYM (SYM: Symmetry) parameter exhibits two to a certain degree undesired properties:

[0088] 1. The peak of the magnitude response shifts in frequency causing an un-desirable change of tonal "center of gravity" when operating the Symmetry parameter. This is due to the fact that in algorithm 1, the corner frequency of a shelf filter is defined as mid-slope frequency, while that of a bell shaped is the frequency where the magnitudes deviate the most from 0 dB.

[0089] 2. At intermediate Symmetry settings, the magnitude may not reach the prescribed gain (G) setting at any frequency. This may not be very intuitive to a user.

[0090] Many users will ignore the above-mentioned properties. According to a further embodiment of the invention, these properties will compensated.

[0091] The first feature may be reduced by mapping the chosen f_c into the pole frequency of all filter symmetries, and thus redefining the meaning of the f_c parameter for the classic shelf type filters (Algorithm 2):

[0092] Given user parameters G in dB, f_c in Hz, Q and symmetry:

$$g=10^{\frac{G}{20}}$$

$$\omega=2\pi f_c$$

$$\omega_z=\omega\cdot g^{-\frac{symmetry}{2}}$$

$$Q_z=Q\cdot g^{|symmetry|-1}$$

-continued

$$g_{correction} = \begin{cases} 1 \text{ if symmetry} \approx 0 \\ \left(\frac{\omega}{\omega_0}\right)^S \text{ otherwise} \end{cases}$$

$$H(s) \equiv g_{correction} \frac{s^2 + \frac{\omega_z}{Q_z} s + \omega_z^2}{s^2 + \frac{\omega_0}{Q} s + \omega^2}$$

if $G < 0$: $H(s) \equiv H(s)^{-1}$

[0093] The above described mapping may be regarded as a frequency compensation of Symmetry parameter invoked equalizer curve modification, when compared to conventional understanding of the corner frequency.

[0094] Evidently, several other more or less intuitive compensations may be applied.

[0095] **FIG. 3***a* illustrate the functioning of Symmetry parameter with constant pole frequency as described above.

[0096] The second property can be reduced by modifying the Gain parameter, the first order numerator coefficient of H(s) when G>0 or the first order denominator coefficient when G<0 according to some empirical function. Note however, the meaningful relationship between the asymptotic gain and the Symmetry setting in **FIGS. 2 and 3**:

$G_{asymptote} \equiv$ symmetry/G, both gains in dB

[0097] The gain compensation may be obtained according to several different approaches if desired. One approach may be that of fixing the asymptotic values (by gain compensation of the resulting filter) of the gain at low frequencies or at high frequencies.

[0098] Another approach would be fixing the gain or attenuation peak at a certain value.

[0099] **FIG. 3***b* illustrates a gain compensation applied for the purpose of equaling the maximum gain obtained at or near the corner frequency. It should be stressed, as stated above, that several other manual or automatic compensation techniques may be applied, both with respect to gain and the corner frequency in order to fit the users expectations with respect to the development of the gain and the frequency when modifying the user adjustable parameters. One of several examples of such may for example be a combination of the above described frequency and gain compensation.

[0100] Such techniques may also imply empirically established compensations.

Invertibility

[0101] **FIG. 6***a* and **FIG. 6***b* illustrate the possibilities and advantages of the herein referred to invertibility of the parametric equalizer according to an embodiment of the invention. In **FIG. 6***a*, User Parameter Settings UPS may be adjusted by a user. Such settings may, according to an embodiment of the invention comprise gain, corner frequency, Q and Symmetry.

[0102] The parameters control suitable hardware means (not shown)

[0103] The adjustable settings may then, in a suitable way be transformed into filter coefficient setting FCS, e.g. coefficient of a biquad filter, analog or digital.

[0104] In **FIG. 6***b*, however, au initial set of Filter Coefficient Settings iFCS is applied as initial coefficient settings of applied filter. These settings may e.g. be retrieved from a bank of settings available to the user. Such initial settings may for example be established on the basis of complex filter design algorithms or they may represent for example settings of preferred filters, earlier tested and approved by the user.

[0105] The settings may then, due to the invertibility of the applied parameters settings and the corresponding filter settings, be converted into corresponding initial User Parameter Settings, iUPS. These settings may then be fine-tuned or modified by the user, by means of his preferred tuning means, the parametric equalizer according to the invention, as illustrated in **FIG. 6***a*.

[0106] This invertibilty-feature is in particular an advantage in relation to audio signal processing due to the fact that the input signals, such as voice or instruments, typically varies quite significantly, thereby requiring individual filter settings, not only with respect to variation of sound, but sometimes also with respect for the rendering "room". Due to the fact that such tuning has to be performed in the parameter domain, it is a significant advantage according to an embodiment of the invention, that filters established on the basis of coefficient settings (e.g. by a filter design program in the coefficient domain) may be presented to the user in the parameter domain.

[0107] According to an embodiment of the invention, the user may now retrieve an initial setting completely described by the available adjustable User Parameter Settings UPS and he may modify the parameters by his preferred adjustment means, the parameter modifications available by means of the parametric equalizer.

[0108] The principle of releasing the last degree of freedom for user adjustment has provided a parametric equalizer, featuring the same benefits obtained by conventional parametric equalizer with respect to easy and flexible tuning together with the possibility of modifying the obtaining equalizer characteristics into several other curve forms than offered until now.

[0109] In principle, the adjustment may be obtained by other types of adjustment parameters than the typical parameters corner frequency, gain and Q.

[0110] In practice, an arbitrary order of the applied filter in the parametric equalizer may be converted into a cascade of biquad filters.

[0111] As long as the equalizer algorithm is not complicated further than for example algorithm 1 or 2. the parametric equalizer according to an embodiment of the invention is invertible, meaning that there exists a unique translation from filter coefficients back to parameters.

[0112] Invertibility may also be expressed as the ability to map a continuum of the coefficient space "back" into parametric equalizer parameter settings.

[0113] An inverse algorithm is a little more complicated (Algorithm 3):

Given $H(s) = \frac{b_0 s^2 + b_1 s + b_2}{a_0 s^2 + a_1 s + a_2}$

$\omega_z = \sqrt{\frac{b_2}{b_0}}$; $\omega_\infty = \sqrt{\frac{a_2}{a_0}}$

$Q_z = \omega_z \frac{b_0}{b_1}$; $Q = \omega_\infty \frac{a_0}{a_1}$

symmetry = 0

If $\omega \neq \omega_z$: symmetry $= \frac{2}{2 - \frac{\log(Q_z) - \log(Q)}{\log(\omega) - \log(\omega_z)}}$ *Endif* *)

If symmetry < 0 or |symmetry| > 1:

symmetry $= \frac{2}{2 - \frac{\log(Q_z) - \log(Q)}{\log(\omega) - \log(\omega_z)}}$ *Endif* *)

If |symmetry| > 0.5:

$g = \left(\frac{\omega_z}{\omega}\right)^{\frac{2}{symmetry}}$ Else $g = \left(\frac{Q_z}{Q}\right)^{\frac{1}{symmetry}-1}$ *Endif*

If $|symmetry|-1| < 10^{-3}$:

$M_{DC} = 20\log_{10}\left(\frac{b_2}{a_2}\right)$; $M_\infty = 20\log_{10}\left(\frac{b_0}{a_0}\right)$

If ($|M_{DC}| < 10^{-3}$ & symmetry < 0) or
($|M_\infty| < 10^{-3}$ & symmetry < 0):

symmetry = −symmetry, $g = \frac{1}{g}$

Endif

Endif

If $g < 1$: $Q = Q_z$; $f_z = \frac{\omega_z}{2\pi}$ Else: $f_z = \frac{\omega}{2\pi}$ *Endif*

$G = 20\log_{10}(g)$

*). Here log is a logarithm with any base

[0114] The invertibility of the fully parametric EQ opens up another line of application besides the normal EQ.

[0115] A filter block applied according to the invention provided may, if it is strictly minimum-phase and has equal number of poles and zeros, no matter if it is the result of human adjustment or computer optimization be transformed back into a parameter set that makes sense to human beings, and thus enables the human user to gain understanding of—and add further fine-tuning to—the result of such a computerized filter design. This can be quite useful in advanced development systems e.g. for tuning active loudspeakers.

[0116] A strictly minimum phase filter has no zeros in the right-hand half-plane including the jω axis in case of an analogue filter or no zeros on or outside the unit circle in case of a filter.

[0117] It may be appreciated that, for the purpose solely of obtaining the possibility of converting a given filter setting into at least one set of corresponding parameters, setting the number of adjustable parameters should at least be the number of the non-trivial degrees of freedom. In other words, NDOFpar ≧ NDOFcoef, where NDOFpar is the number of adjustable equalizer parameters and NDOFcoef is the number of non-trivial degrees of freedom in the filter transfer function. Most preferably NDOFpar=NDOFcoef.

Analog Implementation

[0118] **FIG. 4** illustrates the block diagram of an analog implementation of an embodiment of the invention.

[0119] A section of the full-parametric EQ can be implemented as an analog state-variable filter, whose block diagram is shown in **FIG. 4**. The "1/s" blocks are integrators, the "w" nodes are internal signals and the "a" and "b" connections represent connections with gains.

[0120] The transfer function of this circuit can be found as follows:

$$w_2 = x - \frac{a_1}{s} w_2 - \frac{a_2}{s^2} w_2 \Leftrightarrow w_2 \left[1 + \frac{a_1}{s} + \frac{a_2}{s^2}\right]$$

$$= x \Leftrightarrow w_2 = \frac{1}{1 + \frac{a_1}{s} + \frac{a_2}{s^2}} x$$

$$= \frac{s^2}{s^2 + a_1 s + a_2} x$$

$$w_1 = \frac{1}{s} w_2 = \frac{s}{s^2 + a_1 s + a_2} x$$

$$w_0 = \frac{1}{s} w_1 = \frac{s}{s^2 + a_1 s + a_2} x$$

$$y = b_2 w_0 + b_1 w_1 + b_0 w_2$$

$$= \frac{b_0 s^2 + b_1 s + b_2}{s^2 + a_1 s + a_2} x$$

$$= b_0 \frac{s^2 + \frac{b_1}{b_0} s + \frac{b_2}{b_0}}{s^2 + a_1 s + a_2} x$$

$$H(s) = \frac{y}{x} = b_0 \frac{s^2 + \frac{b_1}{b_0} s + \frac{b_2}{b_0}}{s^2 + a_1 s + a_2}$$

[0121] This can be built from real-world electronics using four op-amps in the classical state-variable configuration as illustrated in **FIG. 5**.

[0122] **FIG. 5** illustrates the electronic schematics of an analog implementation of the block diagram of **FIG. 4**. The voltages v_1, v_2, v_3 and v_{out} can be calculated from v_i as follows:

$$v_3 = v_2 \frac{R_5}{R_5 + R_{12}} \frac{R_2 + R_4 + R_3}{R_1 + R_3} - v_i \frac{R_2}{R_3} - v_1 \frac{R_2}{R_4};$$

$$v_2 = -v_3 \frac{1}{sR_6C_1}; v_1 = -v_2 \frac{1}{sR_1C_2} = v_3 \frac{1}{s^2 R_6 C_1 R_1 C_2}$$

-continued

$$\Leftrightarrow v_2 = -v_3\frac{1}{sR_6C_1}\frac{R_5}{R_5+R_{12}}\frac{\frac{R_2R_3}{R_2+R_3}}{\frac{R_2R_3}{R_2+R_3}+\frac{R_1R_3}{R_1+R_3}} - v_1\frac{R_2}{R_3} - v_2\frac{1}{s^2R_6C_1R_7C_2}\frac{R_2}{R_1};$$

do

$$\Leftrightarrow v_2\left(\frac{1}{sR_6C_1}\frac{R_5}{R_5+R_{12}}\frac{R_1R_2+R_2R_3+R_1R_3}{R_1R_3}+\frac{1}{s^2R_6C_1R_7C_2}\frac{R_2}{R_1}+1\right) = -v_1\frac{R_2}{R_3};\ \text{do}$$

where the "||" operator is defined as

$$x \parallel y = \left(\frac{x \cdot y}{x+y}\right)$$

or equivalently

$$x_1 \parallel x_2 \parallel \ldots \parallel x_n = \left(\sum_{i=1}^{n} x_i^{-1}\right)^{-1}$$

Common denominator: $s^2R_6C_1(R_5+R_{12})R_1R_3R_7C_2$

$$\Leftrightarrow v_2\left(\frac{sR_7C_2R_5(R_1R_2+R_2R_3+R_1R_3)+(R_5+R_{12})R_1R_3R_7C_2+s^2R_6C_1(R_5+R_{12})R_1R_3R_7C_2}{s^2R_6C_1(R_5+R_{12})R_1R_3R_7C_2}\right) = -v_1\frac{R_2}{R_3};\ \text{do}$$

$$\Leftrightarrow \frac{v_2}{v_1} = -\frac{R_2}{R_3}\frac{s^2R_6C_1(R_5+R_{12})R_1R_3R_7C_2}{s^2R_6C_1(R_5+R_{12})R_1R_3R_7C_2+sR_7C_2R_5(R_1R_2+R_2R_3+R_1R_3)+(R_5+R_{12})R_1R_2}$$

$$= -\frac{R_2}{R_3}\frac{s^2}{s^2+s\frac{R_7C_2R_5(R_1R_2+R_2R_3+R_1R_3)}{R_6C_1(R_5+R_{12})R_1R_3R_7C_2}+\frac{(R_5+R_{12})R_1R_2}{R_6C_1(R_5+R_{12})R_1R_3R_7C_2}}$$

$$= -\frac{R_2}{R_3}\frac{s^2}{s^2+s\frac{R_5(R_1R_2+R_2R_3+R_1R_3)}{R_6C_1(R_5+R_{12})R_1R_3}+\frac{R_2}{R_6C_1R_1R_7C_2}};$$

$$v_2 = -v_3\frac{1}{sR_6C_1};\ v_1 = v_3\frac{1}{s^2R_6C_1R_7C_2}$$

[0123] This leaves us with far more than the necessary 2 degrees of freedom for composing the denominator polynomial:

$$Den(s) = s^2 + s\frac{R_5(R_1R_2+R_2R_3+R_1R_3)}{R_6C_1(R_5+R_{12})R_1R_3} + \frac{R_2}{R_6C_1R_1R_7C_2}$$

$$= s^2 + \frac{\omega}{Q}s + \omega^2$$

[0124] To simplify things we choose $R_1=R_2=R_3=R_5=R_{1235}$ and $R_{12}=2R_{1235}$, so

$$Den(s) = s^2 + s\frac{1}{R_6C_1} + \frac{1}{R_6C_1R_7C_2}$$

and (continuing calculations)

$$\frac{v_3}{v_1} = -\frac{s^2}{s^2+s\frac{1}{R_6C_1}+\frac{1}{R_6C_1R_7C_2}};\ v_2 = -v_3\frac{1}{sR_6C_1};$$

$$v_1 = v_3\frac{1}{s^2R_6C_1R_7C_2} \Rightarrow \frac{v_2}{v_1} = \frac{s\frac{1}{R_6C_1}}{s^2+s\frac{1}{R_6C_1}+\frac{1}{R_6C_1R_7C_2}};$$

$$\frac{v_1}{v_3} = -\frac{\frac{1}{R_6C_1R_7C_2}}{s^2+s\frac{1}{R_6C_1}+\frac{1}{R_6C_1R_7C_2}}$$

[0125] So the 3 signals v1, v2 and v3 are LP, BP and HP filtered versions of the input with pass band gains of −1,1 and −1 respectively. Note that these transfer functions are independent of the chosen R_{1235}, which may be chosen arbitrarily to R_{1235}=10 kΩ, for instance. The components determining the pole positions can be chosen as follows:

$$Z_{typ} = 10k\Omega$$

$$C_1 = \text{round\_to\_nearest}\left(\frac{Q}{\omega Z_{typ}}\right)$$

$$\frac{\omega}{Q} = \frac{1}{R_6C_1} \Leftrightarrow R_6 = \frac{Q}{\omega C_1}$$

$$C_2 = \text{round\_to\_nearest}\left(\frac{1}{Z_{typ}C_1\omega^2}\right)$$

$$\omega^2 = \frac{1}{R_6C_1R_7C_2} \Leftrightarrow R_7 = \frac{1}{\omega^2R_6C_1C_2}$$

[0126] Now combining the 3 signals in the summing amplifier (**U4 in FIG. 5**), creates the numerator of the EQ's transfer function:

$$v_{out} = -\frac{R_{11}}{R_{10}}v_3 + \frac{R_4}{R_4+R_9}\frac{R_{11}+R_8\,\|\,R_{10}\,\|\,R_{12}}{R_8\,\|\,R_{10}\,\|\,R_{12}}v_2 - \frac{R_{11}}{R_6}v_1$$

$$= \frac{\frac{R_{11}}{R_{10}}s^2 + \frac{R_4}{R_4+R_9}\frac{R_{11}+R_8\,\|\,R_{10}\,\|\,R_{12}}{R_8\,\|\,R_{10}\,\|\,R_{12}}\frac{1}{R_6C_1}s + \frac{R_{11}}{R_6}\frac{1}{R_6C_1R_7C_2}}{s^2 + s\frac{1}{R_6C_1} + \frac{1}{R_6C_1R_7C_2}}v_1$$

$$= \frac{\frac{R_{11}}{R_{10}}s^2 + \frac{R_4}{R_4+R_9}\frac{R_8R_{10}R_{11}+R_8R_{11}R_{12}+R_{10}R_{11}R_{12}+R_8R_{10}R_{12}}{R_8R_{11}R_{12}}\frac{1}{R_6C_1}s + \frac{R_{10}}{R_6}\frac{1}{R_6C_1R_7C_2}}{s^2 + s\frac{1}{R_6C_1} + \frac{1}{R_6C_1R_7C_2}}v_1$$

[0127] Again, there are too many degrees of freedoms to obtain the desired overall EQ transfer function:

$$EQ(s) = \frac{v_{out}}{v_1}$$

$$= \frac{R_{11}}{R_{10}}\frac{s^2 + \frac{R_4}{R_4+R_9}\frac{R_8R_{10}R_{11}+R_8R_{11}R_{13}+R_{10}R_{11}R_{13}+R_8R_{10}R_{13}}{R_8R_{11}R_{13}}\frac{1}{R_6C_1}s + \frac{R_{10}}{R_6}\frac{1}{R_6C_1R_7C_2}}{s^2 + s\frac{1}{R_6C_1} + \frac{1}{R_6C_1R_7C_2}}$$

$$= g_{correction}\frac{s^2 + \frac{\omega_z}{Q_z}s + \omega_z^2}{s^2 + \frac{\omega}{Q}s + \omega^2}$$

[0128] Selecting R_{10} and R_4 to suitable values (e.g. 10 kΩ) the remaining component values are given by:

$$\frac{R_{11}}{R_{10}} = g_{correction} \Leftrightarrow R_{11} = g_{correction}R_{10}$$

$$\frac{R_{10}}{R_8} = \frac{\omega_z^2}{\omega^2} \Leftrightarrow R_8 = R_{10}\frac{\omega^2}{\omega_z^2}$$

$$\frac{R_4}{R_4+R_9}\frac{R_8R_{10}R_{11}+R_8R_{11}R_{13}+R_{10}R_{11}R_{13}+R_8R_{10}R_{13}}{R_8R_{11}R_{13}} = \frac{\omega_z}{\frac{\omega}{Q}} = \frac{\omega_zQ}{\omega Q_z}$$

$$\Leftrightarrow R_4\frac{R_8R_{10}R_{11}+R_8R_{11}R_{13}+R_{10}R_{11}R_{13}+R_8R_{10}R_{13}}{R_8R_{11}R_{13}} = \frac{\omega_zQ}{\omega Q_z}(R_4+R_9)$$

$$\Leftrightarrow \frac{\omega_zQ}{\omega Q_z}R_9 = R_4\left(\frac{R_8R_{10}R_{11}+R_8R_{11}R_{13}+R_{10}R_{11}R_{13}+R_8R_{10}R_{13}}{R_8R_{11}R_{13}} - \frac{\omega_zQ}{\omega Q_z}\right)$$

$$\Leftrightarrow R_9 = \frac{\frac{R_8R_{10}R_{11}+R_8R_{11}R_{13}+R_{10}R_{11}R_{13}+R_8R_{10}R_{13}}{R_8R_{11}R_{13}} - \frac{\omega_zQ}{\omega Q_z}}{\frac{\omega_zQ}{\omega Q_z}}R_4$$

$$= \left(\frac{R_8R_{10}R_{11}+R_8R_{11}R_{13}+R_{10}R_{11}R_{13}+R_8R_{10}R_{13}}{R_8R_{11}R_{13}}\frac{\omega Q_z}{\omega_zQ} - 1\right)R_4$$

[0129] Note that R_9 may become negative. To prevent this from happening within a selected parameter range, we can select a suitably low R_{13} to boost the amplification of the summing amplifier's non-inverting input.

Digital Implementations

[0130] When attempting to make a digital signal processing system work like an analog prototype, like our equalizer, a number of compromises must be made. The discrete-time nature of the digital system causes the frequency representation of digital signals to be limited to the range from 0 Hz to the Nyquist frequency f_{Ny} (half the samplingrate f_s), while in the continuous analog world, the frequency axis continues towards infinity. The mapping of the infinite analog frequency axis onto the finite digital frequency axis can be done in several, imperfect ways.

Direct Implementation by Bilinear Transform

[0131] A computationally convenient method with some virtue is the Bilinear Transform, which maps the entire analog frequency axis (actually the imaginary axis in the complex s-plane) onto the digital frequency axis (actually the unit circle in the complex z-plane), and ensures that stable analog systems are mapped into stable digital systems. The mapping of an infinitely long axis onto a circle of finite circumference is bound to involve some sort of compression or warping. To ensure that the corner frequency of the digital equalizer ends up at the desired value in spite of the warping, it must be pre-warped before doing the design. Unfortunately this only ensures that this one frequency is mapped correctly, the others are still warped, causing a distorted frequency response at high frequencies near f_{Ny}.

[0132] The design of a digital version of the parametric equalizer by bilinear transform requires these steps:

[0133] 1. Prewarp the desired center frequency f_c of the resulting digital filter into an analog design center frequency

$$f_{c,a} = \frac{f_s}{\pi}\tan\left(\frac{\pi}{f_s} f_c\right)$$

[0134] 2. Design the analog EQ (EQ: Equalizer) by the earlier described Algorithm 2

[0135] 3. Apply the bilinear transform by substituting the complex frequency variable s in the analog EQ transfer function by

$$s = 2 f_s \frac{1 - z^{-1}}{1 + z^{-1}}$$

[0136] 4. Renormalize the digital transfer function t to

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

[0137] Because the bilinear transform is invertible, the invertibility property holds for the digital implementations of the fully parametric equalizer, when the direct implementation by bilinear transform is used.

Implementation by Digital Design

[0138] Do we really need to "design" our digital EQ by some transformation of the analog filter coefficients? Why not use mathematics to approximate the magnitude response of the digital filter directly to that of the analog prototype, or to any other target response for that matter? In simplified terms, this method goes as follows:

[0139] 1. Convert user parameter settings (G,fc,Q,Symmetry) into analog coefficients described above.

[0140] 2. Calculate samples of the analog filter's magnitude response at an appropriate selection of frequencies

[0141] 3. Design a bi-quadratic digital filter to fit the sampled magnitude/frequency points, using general purpose IIR filter design techniques

[0142] The digital design method is much preferable if it can be implemented with sufficient computational efficiency on a product platform. Note that it even supports f_c settings above the Nyquist frequency.

[0143] Since the Implementation by a digital design method in general involves approximate IIR filter design techniques such as least-squares approximation, it may not be invertible, but an inverse approximation may be found, yielding only approximate invertibility. Therefore the Direct Implementation by Bilinear Transform may be the preferable method in cases where exact invertibility is important.

[0144] **FIG. 7** illustrates a principle design of the filtering means FM of an embodiment of the invention.

[0145] The illustrated filtering means FM of a parametric equalizer according to an embodiment of the invention comprises a number of filter blocks FIB, here four.

[0146] The filter blocks FIB may be cascaded to form one resulting filtering means.

[0147] The individual filtering blocks FIB may according to a preferred embodiment of the invention preferably each comprise a biquad filter

[0148] Each of the illustrated filtering blocks FIB is moreover individually controlled by a filtering block user interface means FIBUIM. In other words, each of the illustrated filter blocks may be controlled by a user in the parameter domain by means of for example the parameters corner frequency (fc), Shape (Q), gain (G) and symmetry (SYM). Again, in this context Gain is expressed conventionally as boost/attenuation characteristic while the overall gain is referred to as the general volume setting of the individual filter block. The overall gain may typically be shared between all cascaded filters as a common volume setting.

[0149] On other words, a control parameter other than the above described four may be the global or overall gain, which may be applied to the individual filters or more likely as one shared trivial volume control.

[0150] It should of course be noted that the number of filtering blocks of a device according to the invention in principle may vary from one to for example hundreds.

[0151] Typically, a relatively low number of filter blocks FIB which may be cascaded is preferred, e.g. 3 to 8.

[0152] The resulting and/or the individual filter curve settings may be illustrated on one or more displays.

[0153] It should moreover be noted that the applied filter blocks comprise biquad filters. However, other filter types of smaller or larger order may be applied if suitable.

1. Parametric equalizer comprising

filtering means (FM), user interface means (UIM), audio signal input means and audio signal output means,

said filtering means comprising at least one filter block (FIB)

said user interface means (UIM) comprising means for adjustment of parameters: corner frequency (fc), shape (Q) and gain (G),

said parametric equalizer comprising further means for adjusting a symmetry parameter independent to the other user parameters, which may be continuously varied in order to provide a smooth transition between low-shelf, bell-shaped and high-shelf filter characteristic of said at least one filter block (FIB).

2. Parametric equalizer according to claim 1, wherein

said user interface means (UIM) comprises a further symmetry adjustment parameter (SYM) for establishing a variable symmetry of the magnitude response of said at least one filter block (FIB),

said user interface means is mapped by means of coefficient adjustment algorithms into filter coefficient set-

tings (FCS) of the at least one filter block (FIB), which when established reflects the adjustment of the user interface means (UIM)

said further adjustment parameter (SYM) provides a filter coefficient setting (FCS) comprising a combined adjustment of at least one zero frequency, pole frequency, zero Q and pole Q of the magnitude response of said at least one filter block.

3. Parametric equalizer according to claim 1, wherein said user interface means facilitates adjustment of corner frequency (fc), Shape (Q), gain and symmetry.

4. Parametric equalizer according to claim 2, wherein said filter coefficient settings (FCS) comprise digital coefficients.

5. Parametric equalizer according to claim 2, wherein said filter coefficient settings (FCS) comprise analogue values established by means of adjustable or selectable filter components of said at least one filtering means.

6. Parametric equalizer according to claim 1, wherein said filtering means comprises less than twenty individually adjustable filter blocks (FIB).

7. Parametric equalizer according to claim 1, wherein at least one of said filtering blocks comprise a biquatic filter.

8. Parametric equalizer according to claim 1, wherein said parametric equalizer comprises at least one, cascaded biquadratic filters blocks (FIB) .

9. Parametric equalizer according to claim 1, wherein said filtering means is analogously implemented.

10. Parametric equalizer according to claim 1, wherein said filtering means is digitally implemented.

11. Parametric equalizer according to claim 2, wherein said filtering means comprises gain compensation means adapted for compensation of alteration of the filtering block gain invoked by a changed setting of the further adjustment parameter.

12. Parametric equalizer according to claim 2, wherein said filtering means comprises corner frequency compensation means adapted for compensation of alteration of the corner frequency of the filtering block invoked by a changed setting of the further adjustment parameter.

13. Parametric equalizer according to claim 2, wherein said user interface provides at least four different asymmetries of filter setting at least in part of the frequency range.

14. Parametric equalizer according to claim 2, wherein said further adjustment parameter (SYM) enables the user to gradually transform the filter block (FIB) between a low-shelf and a high-shelf filter characteristic.

15. Parametric equalizer according to claim 2, wherein said further adjustment parameter (SYM) enables the user to gradually transform the filter block (FIB) from a low-shelf into a bell-shape and further into a high-shelf, thus defining at least one more than said three standard filter types.

16. Parametric equalizer according to claim 1, wherein a number of said adjustment parameters corresponds to a number of non-trivial degrees of freedom of the at least one filter block (FIB).

17. Parametric equalizer according to claims **7**, wherein a number of said adjustment parameters is at least a number of non-trivial degrees of freedom of the at least biquad filter block (FIB) times the number of filter blocks (FIB) of said filtering means.

18. Parametric equalizer according to claim 8, wherein a number of non-trivial degrees of freedom of each of a number of said cascaded filter blocks is at least four.

19. Parametric equalizer according to claim 2, wherein the symmetry parameter may be set by means of the user interface to at least four different values.

20. Parametric equalizer according to claim 1, wherein the adjustment parameters are converted into filter coefficient settings (FCS) triggered by setting of the adjustment parameters by the user.

21. Parametric equalizer according to claim 20, wherein the conversion of adjustment parameters into filter coefficient settings is invertible.

22. Parametric equalizer according to claim 1, wherein NDOFpar≧NDOFcoef, where NDOFpar is the number of adjustable equalizer parameters and NDOFcoef is the number of non-trivial degrees of freedom (fc, G, Q, Sym).

23. Parametric equalizer according to claim 1, wherein given filter coefficient settings may be converted into corresponding adjustment parameters.

24-26. (canceled)

* * * * *

(x) RELATED PROCEEDINGS APPENDIX

None.